2020-09

SIMULATION MODEL OF SURFACE WARFARE OFFICER TRAINING PIPELINE TO MINIMIZE FRICTION

Vranich, Joseph

Monterey, CA; Naval Postgraduate School

http://hdl.handle.net/10945/66042

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

SIMULATION MODEL OF SURFACE WARFARE OFFICER TRAINING PIPELINE TO MINIMIZE FRICTION

by

Joseph Vranich

September 2020

Thesis Advisor:	Roberto Szechtman
Second Reader:	Chad W. Seagren

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2020	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE SIMULATION MODEL OF SURFACE WARFARE OFFICER TRAINING PIPELINE TO MINIMIZE FRICTION	**5. FUNDING NUMBERS**
6. AUTHOR(S) Joseph Vranich	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

Surface warfare officers (SWO) currently attend Basic Division Officer Courses (BDOC) and Junior Officer of the Deck (JOOD) c ourses prior to their first division officer (DIVO) tour. Due to the commissioning schedule, the schedule of these different courses and the number of seats available, there is typically a large wait time for SWOs during different times of the year between classes and before they enter the fleet. These wait times are costly and lower the military readiness of the fleet.

With a growing naval force, minimizing the friction that SWOs experience can produce more department heads in order to man the larger fleet by training more junior officers today. Current analysis is done in Excel and calculations must be repeated for different scenarios. These scenarios include different arrival rates and distributions, service rates, and population sizes. Using Simio and Python, simulation models of a queuing network can be created in order to find different queue lengths and waiting times automatically using historic commissioning rates and schedules. Once a proof-of-concept model has been created, the model inputs can be changed to match predicted inputs for a growing naval force. Then allocations of training schedules and capacities can be recommended in order to optimize the SWO training pipeline.

14. SUBJECT TERMS Surface Warfare Officer, friction, Simio, Python	**15. NUMBER OF PAGES** 81
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500 — Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

SIMULATION MODEL OF SURFACE WARFARE OFFICER TRAINING PIPELINE TO MINIMIZE FRICTION

Joseph Vranich
Lieutenant, United States Navy
BS, California State University - Chico, 2011

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 2020

Approved by: Roberto Szechtman
Advisor

Chad W. Seagren
Second Reader

W. Matthew Carlyle
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Surface warfare officers (SWO) currently attend Basic Division Officer Courses (BDOC) and Junior Officer of the Deck (JOOD) courses prior to their first Division Officer (DIVO) tour. Due to the commissioning schedule, the schedule of these different courses and the number of seats available, there is typically a large wait time for SWOs during different times of the year between classes and before they enter the fleet. These wait times are costly and lower the military readiness of the fleet.

With a growing naval force, minimizing the friction that SWOs experience can produce more department heads in order to man the larger fleet by training more junior officers today. Current analysis is done in Excel and calculations must be repeated for different scenarios. These scenarios include different arrival rates and distributions, service rates, and population sizes. Using Simio and Python, simulation models of a queuing network can be created in order to find different queue lengths and waiting times automatically using historic commissioning rates and schedules. Once a proof-of-concept model has been created, the model inputs can be changed to match predicted inputs for a growing naval force. Then allocations of training schedules and capacities can be recommended in order to optimize the SWO training pipeline.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AB	Agent Based
ADF	Australian Defense Force
ADOC	Advanced Division Officer Course
BDOC	Basic Division Officer Course
BST	Billet Specific Training
CO	Commanding Officer
COVE	Conning Officer Virtual Environment
CI	Confidence Interval
CIC	Combat Information Center
CRT	Comprehensive Review Team
CSV	Comma Separated Value
DES	Discrete Event Simulation
DIVO	Division Officer
DOTMLPF	Doctrine, Organization, Training, Material, Leadership, and Education, Personnel, and Facilities
DH	Department Head
DLL	Dynamic-Link Library
EG	Event Graph
EOOW	Engineering Officer of the Watch
JOOD	Junior Officer of the Deck
MAT	Marines Awaiting Training
MCCES	Marine Corps Communication-Electronics School
NSF	Naval Surface Force
NSTC	Naval Service Training Command
OCS	Officer Candidacy School
OJT	On-the-Job Training
PAN	Process Analyzer Function
PQS	Personnel Qualification System
PST	Platform Specific Training
ROTC	Reserve Officer Training Corps

RROC	Readiness Reform and Oversight Council
SIMIO	Simulation Modeling Framework Based on Intelligent Objects
SWO	Surface Warfare Officer
SWOS	Surface Warfare Officer School
USFF	United States Fleet Forces
USNA	United States Naval Academy
XO	Executive Officer

EXECUTIVE SUMMARY

There are approximately 9,000 Surface Warfare Officers (SWOs) in the U.S. Navy at any given period. These officers undergo the same training at the beginning of their careers. They start off with a Basic Division Officer Course (BDOC) within six months of commission, followed by a Junior Officer of the Deck (JOOD) training course. JOOD is relatively new and when coupled with BDOC, is meant to provide hundreds of simulated standard watch hours for individual and team environments. BDOC courses are in San Diego and Norfolk. JOOD courses are in San Diego, Norfolk, and Newport.

The purpose of adding JOOD to the initial training pipeline is to improve the overall seamanship and navigation quality of the typical SWO while preparing them to handle extreme emergency situations. The improvement and implementation of bridge trainers and training evaluations that students see at JOOD is a result of multiple mishaps in 2017 that saw a combined loss of 17 sailors. These mishaps were all due, in part, to failing to handle extreme situations, watchstanders failing to comply with procedures, and watchteams not working as an effective team.

Adding the JOOD course to the training pipeline prior to a SWO's first Division Officer (DIVO) tour will add friction to the training pipeline. The goal of this thesis is to model the first portion of the SWO training pipeline to include initial commission, BDOC, and JOOD courses to measure the overall friction a SWO experiences prior to their assignment to the fleet. This model should be user friendly and flexible so that senior leadership can observe how changes to the initial pipeline impact certain metrics such as average wait time and number of SWOs waiting at all stages of this initial training. Upon completion of the model, the schedule can be optimized to minimize the overall wait time that a typical SWO experiences while potentially graduating more SWOs per year. Inputs for each model is based on historical data for Eastern and Western accessions and existing class schedules provided by PERS-41. Eastern and Western accessions are classified based on the final platform to which a SWO will report.

The entity-based simulation program Simio and a Discrete Event Simulation (DES) package, DESpy, in Python were both used to model the initial training pipeline. Simio and Python provide a discrete time simulation. The Simio simulation was chosen initially due to its user-friendly interface and visualization capabilities. Python was eventually brought in instead, due to it being open source and more flexible; whereas Simio requires licensing agreements that could provide complications for Navy-wide distribution. Both programs can process large numbers of objects that can interact with servers that follow a strict calendar schedule. They also work well with simple spreadsheet programs such as Excel that senior leadership can manipulate and provide as inputs to the simulations. Simio is the short-term model that provided the initial analysis and recommendations for the training pipeline. Python is the long-term model that has comparative results to Simio and provides a foundation for future work for either improving the existing DIVO model or other sections of the SWO training pipeline, such as Department Head (DH) training.

Both models follow similar logic. Western accessions attend BDOC in San Diego while Eastern accessions attend BDOC in Norfolk. Accessions can attend BDOC on the opposite coast if seats are available and if BDOC on the same coast is at capacity. Students that graduate BDOC in San Diego also attend JOOD in San Diego while students that graduate BDOC in Norfolk will attend JOOD in Norfolk. JOOD in Newport will take the students who have been waiting the longest.

The results of the Simio model provide an average wait time 191.78 ± 2.79 days for Eastern accessions and 216.71 ± 3.29 days for Western accessions based on 50 degrees of freedom and a 95% confidence interval. A maximum of 1,164 SWOs can make it through training for one year of steady-state simulation time based on the current schedule and classroom capacities. Steady state is assumed to be reached after about 150 days of warmup period to accommodate for zero students being in the system at the beginning of the simulation. Average number of students waiting is broken down by month, with most students waiting in the summer months between May and August. This makes sense considering the majority of students that enter the system commission in May. There are five BDOC courses per year in San Diego and Norfolk. Only three of these classes are fully utilized by students, while class utilization rate drops to between 20% and 40% for the other classes. Norfolk and San Diego have 14 classes pear year while Newport has 10 classes. Two classes in Norfolk and three classes in San Diego are completely unused throughout the year while one class in Newport is completely unused. The remaining courses are at or near full capacity.

Compressing the JOOD schedule is recommended to optimize the number of SWOs that complete the DIVO training and minimize the average time waiting. Compressing a schedule involves minimizing time in between classes to fit one more JOOD class in per location per year. Compressing only Norfolk or San Diego would result in 18 ± 3 days less wait time per SWO while allowing 1,194 SWOs maximum to complete their training per year. Compressing Norfolk and San Diego would result in 27 ± 3 days less while allowing 1,224 SWOs maximum. Compressing all JOOD schedules would result in 28 ± 3 days less and 1,248 SWOs maximum. Based on historical data and the number of expected SWO candidates that commission each year, compressing either Norfolk or San Diego JOOD is recommended. If more commissions happen in a year, compressing both Norfolk and San Diego JOOD is recommended.

These recommendations will lower the amount of friction a SWO experiences each year in the DIVO stage of their training. The DH stage of their training has much more opportunity for improvement. In the DH stage, SWOs attend training based on billet- and platform-specific requirements, creating more complicated modeling requirements. Modeling and optimizing DH training would potentially reduce friction substantially more than any recommended change in DIVO training.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

There are several people that I am grateful to for their support, directly or indirectly, with this project. Thank you first and foremost to my advisor, Dr. Roberto Szechtman. Initial struggles getting this thesis off the ground while receiving several offers to pursue other projects left me with low motivation in the beginning. He remained supportive of my decision while maintaining direction for the project, which ultimately led to my choice of continuing with this topic.

Thank you to my second reader, Dr. Chad Seagren. His initial research and communication with PERS-41 created the momentum needed to begin preliminary modeling. His review and input also helped create several metrics that did not exist in the original analysis.

Thank you to Josh Perry for his initial help with the model. This was meant to be a project shared between the two of us in the beginning before he pursued other opportunities. Simio was a new program for each of us and he made several initial breakthroughs that propelled the project forward. His work on the initial Simio model created a strong foundation for the final model.

Finally, thank you to my wife, Kelly. Having a child halfway through graduate school and at the beginning of my thesis work may not have been the best timing, but her encouragement in allowing me to focus on my work throughout the last year while giving me peace of mind knowing our daughter, Hailey, was in good hands was the best support I could ask for. I love you both.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

The discussion in this thesis opens with background information on Surface Warfare Officers (SWOs) and their training pipeline. This is followed by discussion on how the training pipeline has changed and thus what is the problem to be solved in this thesis. A brief introduction to the models used to solve this problem is provided followed by a roadmap for this report.

A. BACKGROUND

Throughout the world, the U.S. Navy has approximately 9,000 SWOs stationed in 10 homeports. These SWOs operate a wide "variety of ship types such as Aircraft Carriers, Destroyers, Cruisers, Mine Countermeasure ships, Littoral Combat Ships, and Amphibious ships" (Department of the Navy 2017, page 46). In order to operate these ships, there are courses that an SWO must take at the beginning of their training pipeline, prior to their first Division Officer (DIVO) tour. The first course attended by a SWO after commissioning is Basic Division Officer Course (BDOC), which is meant to be taken within six months of commission. The second course, Junior Officer of the Deck (JOOD) course, commenced in May 2019 and is scheduled for full implementation in 2021 (Department of the Navy 2018). BDOC is 9-week course of instruction with classes located in Norfolk and San Diego. The course is designed to provide individual hands-on training as well as in-class instruction and the use of technology to simulate every class of U.S. Navy ship. The instruction provided is in the areas of "division officer fundamentals, maritime warfare, engineering, leadership, and damage control" (Department of the Navy 2017). JOOD is meant to be taken upon the completion of BDOC. Class locations include Norfolk, San Diego, and Newport. While BDOC emphasizes the individual training, the JOOD course emphasizes to the SWO candidates how to work as part of a bridge team. These team simulations are applied to "some of the world's busiest shipping lanes and traffic separation schemes" (Commander, Naval Surface Force, U. S. Pacific Fleet 2019). The first four-week JOOD class graduated in July 2019.

The addition of the JOOD courses are a response to major at-sea mishaps that have happened in the past decade. Four of these mishaps, three of them collisions, occurred in 2017. The two most recent mishaps were collisions involving U.S. destroyers that resulted in the combined loss of 17 U.S. Sailors (Department of the Navy 2017). These issues have sparked an effort to update the curricula for SWOs as well as Quartermasters, Operations Specialists, and Electronic Technicians. The JOOD course is specifically aimed at improving the individual skills training in seamanship and navigation for SWO candidates. In response to common themes found in the collisions of 2017, the team exercises in the JOOD simulation provide actionable experiences that implement high-density traffic, emergency, and extremis situations. The Navy Readiness Reform and Oversight Council (RROC) provided recommendations based on the comprehensive reviews that will be discussed further in the literature review.

BDOC and JOOD training are ideally given prior to the SWO's first DIVO tour. However, due to scheduling and the class capacities, there is a lot of friction for SWOs between the commission date and the first fleet tour. Friction is defined as the amount of wait time that the student experiences throughout the training process. The addition of the JOOD course adds to the amount of friction a SWO will experience in their training pipeline. The course, however, when combined with BDOC provides demonstrable experience that translates to fleet experience. Captain Chris Alexander, Commanding Officer (CO) of SWO school, has stated that the 130 hours that these officers spend in the simulators translates to the equivalent of more than 40 standard watches by the time they report to their respective ships (Commander, Naval Surface Force, U. S. Pacific Fleet 2019). A common practice for a SWO candidate that has completed BDOC is to report to their ship and continue their training while waiting to class up to JOOD. However, due to the operational schedule of that ship, they may not be able to attend the earliest possible JOOD course. The Navy also plans on greatly increasing the ship count, starting in the late 2020s (Cancian 2019). Modeling the beginning of the training pipeline to increase the efficiency can have significant long-term payoffs in ensuring the ships are properly manned.

The SWO Career Path is outlined in Figure 1. The top left of the figure shows the beginning of their path. It shows that BDOC, JOOD and Billet Specific Training (BST) come before their first DIVO tour. The requirement to take BST, as well as the duration, which can last anywhere between 0 to 18 weeks, is dependent on the billet. The number of officers required to take BST are relatively low and will not be considered in the analysis. The training prior to the first DIVO tour, according to the timeline, completes shortly before the one-year mark. The model in this thesis simulates this initial DIVO training in order to assess how much friction the SWO will experience during this timeframe.

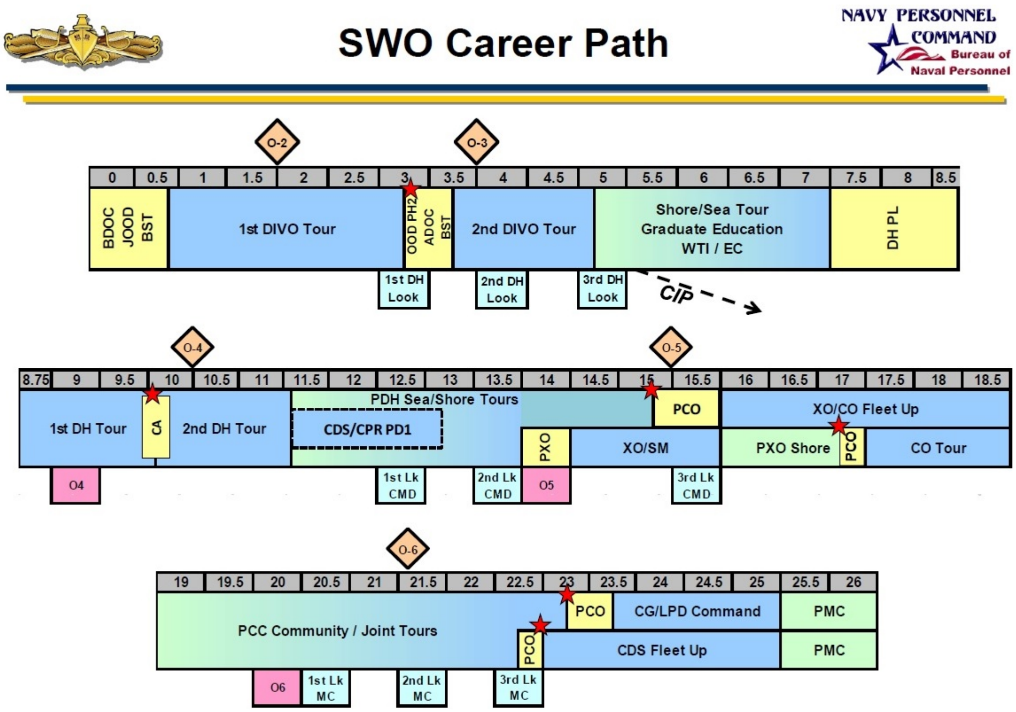


Figure 1. SWO Career Path. Source: Trinque (2019).

B. THE PROBLEM

The goal of this thesis is to provide a flexible model for senior leadership to see how BDOC and JOOD wait times are correlated and how any potential changes would impact friction throughout this section of the training pipeline. The wait schedules and numbers provided by this thesis are best-case scenarios. That is, it prioritized each student by accession date in order to minimize completion time for both BDOC and JOOD courses.

1. Short-Term Model: Simio

To create the model, this project required an entity-based simulation program called Simulation Modeling Framework Based on Intelligent Objects (Simio).[1] Using historic commission rates and set class schedules and class sizes, a full simulation was built in order to observe monthly trends in friction throughout a one-year time period. One aspect of this model is the ability to change the inputs to the model quickly by creating a link between the simulation and Excel. This link allows for a swift import of data provided by PERS-41. Any desired changes leading to a sensitivity analysis can be done quickly to observe the overall impact to the training timeline.

Another aspect of this model is the ability to run experiments. Once all the operational constraints have been set, the simulation can be run a specified number of times independently in order to perform statistical analysis on the outputs. Changes in inputs and the results of these changes as well as conclusions and recommendations are provided in Chapters IV and V.

2. Long-Term Model: Python

Another goal of this thesis is to provide another model using a Discrete Event Simulation (DES) package (DESpy)[2] that implements Event Graphs in Python. Simio, in conjunction with Excel and R, provide the results and analysis later in this report, but Python will be used for future work with the continuation of this model. Python, another object-oriented programming language, has the advantage of being an open source program, unlike Simio, which requires purchase and licensing, a complication for Navy-wide distribution. Another benefit of DESpy is that the user has full control over the implementation, whereas the building blocks used in Simio have a specification that cannot be changed by the user. The DESpy package implements the design of event graphs within the Python language. These event graphs, discussed in further detail in Chapter II, provide the framework for the logic followed by the DESpy package.

---

[1] For more information, see https://www.simio.com/resources/white-papers/Introduction-to-Simio/

[2] For more information, see https://pypi.org/project/DESpy/

C. SUMMARY

The advantages of using Python over Simio are not obvious with the initial DIVO model. Both programs have the aforementioned ability to work with spreadsheet programs to bring in data, create tally statistics, perform sensitivity analysis, and run experiments for stochastic evaluation. The primary differences become apparent with the level of complexity of the model. Simio provides easy implementation to basic models with minimal logic, which is the case for the DIVO portion of the SWO training. However, future work beyond this thesis will include DH training. DH training is a much more complex section of the pipeline. Currently, the DH training for an SWO requires an initial DH school training. This is followed by the SWO attending training specific to their platform and training specific to their billet, adding a level of complexity that will be better handled using Python. Each of 14 classes of Surface Force ships have lists of billets associated with them, creating a large number of combinations for which SWO attends what training. The complexity of this model requires more control than a program like Simio can offer. A Simio model has been created to provide proof of concept and quick turn-around to PERS-41 for analysis. The results of this model will be used as a check for the DIVO portion of the Python model for further proof of concept. Future models of the SWO pipeline beyond DIVO training will be done in Python. The event graph implementation of this model is discussed in Chapter III.

Chapter II discusses the history and background for the creation of JOOD training in the SWO pipeline and the technical concepts used in building the model. Chapter III outlines the methodology behind the model. Chapter IV summarizes the results of the model using data provided by the Navy as well as a sensitivity analysis when changing the schedule to minimize friction. Chapter V gives a conclusion, recommendations moving forward, and future work for this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

II. LITERATURE REVIEW

Part I of this literature review explores the research that describes the historical readiness of the Navy and the consequences of naval decisions. This research led to the introduction of Junior Officer of the Deck courses being added to the training pipeline. Chapter IV explores the analysis of the SWO training pipeline with this additional course. Part II discusses related works that utilize discrete models to solve similar training issues to this thesis. Part III explores the technical concepts used in building the model such as queueing models and event graphs.

A. PART I: HISTORY AND BACKGROUND

1. Comprehensive Review of Recent Force Incidents

In 2017 there were three collisions and one grounding of U.S. Navy ships that resulted in the combined loss of 17 U.S. Sailors (Department of the Navy 2017). All the individual instances have primary causes and contributing factors, but every mishap has certain aspects in common. Paragraph 1.3.1 states:

> In each case, bridge and CIC watchstanders did not maintain situation awareness and recognize that a significant error chain was in motion. Once confronted with an extreme situation, watchstander actions failed to comply with procedures as outlined in governing directives, as well as common customs of service, such as standard commands. Additionally, watchstanders did not take emergency actions, to include sounding alarms, signaling to the other ship, or warning the crew. In every mishap, departures from procedures or approved customary practices were deemed to have directly contributed to the mishap. (Department of the Navy 2017, page 16)

Another problem is the bridge and CIC teams did not work with each other as an effective team to solve the problems. These four mishaps are part of 12 major at-sea mishaps in the past decade, with some commonalities that include deficiencies in fundamentals, operational safety, teamwork, assessment, and culture (Department of the Navy 2017). Each involved an investigation that led to lessons learned and recommendations in order to prevent singular causes to each specific situation, but they did not assess the institutional Doctrine, Organization, Training, Material, Leadership and

Education, Personnel, and Facilities (DOTMLPF) that can externally add to these mishaps. The Comprehensive Review Team (CRT) that compiled the USFF Comprehensive Review determined several areas that required change to the SWO training culture. These areas include too much emphasis placed upon on-the-job training (OJT) and breadth of experience in different professional competencies, not enough emphasis being placed on reinforcing superior seamanship in navigation, and SWO training not including an "assessment of seamanship and navigation knowledge and skills, including in emergency extremis scenarios, in preparation for their next milestone assignment" (Department of the Navy 2017).

The BDOC course that SWOs attend teach about half of the fundamentals included on the Surface Warfare Officer Personnel Qualification System (PQS). The rest is provided by a ship's training program. The partial curriculum is due to the length of BDOC, which is on average, only 60 days. Due to the initial training and the number of prerequisites required by each SWO, as well as the requirement that the surface warfare qualification must be complete within 18 months, the CRT noted that there is a larger value placed on qualification, as opposed to experience and proficiency. Different accession sources, such as the U.S. Naval Academy (USNA), Officer Candidacy School (OCS), and Naval Service Training Command (NSTC), also are inconsistent in providing foundation standards for core competencies. Operation schedules also shift from ship to ship. Depending on the port, a SWO candidate may gain significant exposure to training and certification, operation deployments, or shipyard maintenance. Therefore, experience among each individual SWO may vary greatly.

Lack of teamwork and inability to perform in emergency situations are factors that contributed to the creation of the JOOD course. As described in the introduction, students of this course are provided the opportunity to work as a team in simulations that mimic congested areas and in situation of extremis. The increased team training and improved Bridge trainers have the goal of increasing the SWO's overall functionality.

2. U.S. Navy Strategic Readiness Review

The U.S. Navy has maintained a relatively consistent number of ships on deployment while having fewer ships available (Department of the Navy 2017). This means that the percentage of the fleet deployed has risen to about double. This has been the trend since roughly 1989 (Department of the Navy 2017). The result has increased the operating rhythm of ships, but lowered the opportunity to train, perform readiness certifications, and maintenance. The “must-do” culture was embraced for immediate mission accomplishment at the cost of long-term readiness and capability.

The must-do culture led to pushing the certifications and readiness to the fleet. This caused more stress on crews through longer workdays and workloads. Given these conditions and other decisions that added more risk factors over several years, the review revealed that “the Navy was off course” (Department of the Navy 2017). The Navy being off course was emphasized by the ship accidents in 2017, four accidents overall, two leading to loss of life.

Training has also been pushed to the side in favor of more reliance on fleet operations. In 2003, a 16-week Surface Warfare Officer School (SWOS) basic course was discontinued in favor of more on-the-job training (Department of the Navy 2017). The officers who missed out on this training are now serving as executive officers (XOs) and commanding officers (COs). One recommendation provided by this report that is relevant to the training pipeline being modeled in this thesis is, “Restructure officer career paths, particularly for surface warfare officers, to refocus on mastering skills crucial to the Navy” (Department of the Navy 2017). Another is to complete relevant simulator training scenarios in order to retain watchstanding qualifications.

Incorporating JOOD into the training pipeline has already occurred in order to create a senior leadership that not only has high on-the-job training, but also dedicated training experience. Modeling the initial training pipeline will provide insight into not only the timeline of SWOs in steady state conditions, but also how changes to rising commission rates will impact things like friction and classroom capacities. Observing potential changes in training schedules and how they can impact future trainings, such as Department Head

(DH) school, is an overall goal. However, based on time constraints, modeling Department Head training in conjunction with Division Officer training will be categorized as future work for this thesis.

B. PART II: RELATED WORK

1. Aircrew Manpower Supply Modeling

The Australian Defense Force (ADF) looks for ways to improve the efficiency on their training and operations. These changes range from small changes to policy to large improvements to infrastructure. The authors of this paper use an Agent Based (AB) DES methodology to simulate the training pipeline for manpower planning and supply. The resources, or students, perform activities as well as conduct human interactions that the model captures in order to determine how individual behavior determines activity progress (Nguyen et al. 2017). The purpose of the training pipeline simulator is to provide long term support for planning manpower needs of the ADF. The training pipeline in this study spans multiple training schools all the way through graduation, where the progression of each student is impacted by the instructor, availability of resources such as aircrafts, pass rates, policy, and other operational requirements (Nguyen et al. 2017).

The model has a transient nature where the user can reconfigure the model at different checkpoints of the model. One example is once one school is completed by an entity, the school is entity is moved and the pass rate is changed to a different mean for the next school. The distribution of the pass rate can also be changed if desired. This is an important step because everyone in the simulation is distinguishable. Every entity has their own training route and record because of their unique career path. Every checkpoint, or node, in the simulation has its own decision logic that characterizes the policy of the training institution that the students attends. The different policies that ADF wants to explore are to conduct a “what-if” analysis that is incorporated at each node.

The DES training model in this thesis does not take an AB approach and does not have a human interaction component. The only distinguishable factor of the different entities in the simulation is whether the entity is an Eastern or Western accession. There are also no factors for instructor performance and other individual requirements. The requirements are the same for all students and there are no shortages of resources. The models in this thesis have a more static nature to them based on how SWOs complete their DIVO training. The primary policy this thesis is concerned with is what changes to the training schedule can graduate at least the same number of SWOs while reducing the friction as much as possible. The foundation of the DES structure where the entities travel from node to node from commission to graduation is very similar. The distinguishability of the entities and the career paths that they follow would be an important factor in the DH training for future work considerations. At the DIVO stage of training, the training is identical regardless of platform or billet. At the DH stage, a SWO will attend a class based on what billet and platform they are assigned to.

2. Discrete Event Simulation to Examine Marine Training

The Marine Corps Communication-Electronics School (MCCES) in Twentynine Palms, CA, prepares Marines to operate network systems and various transmission media. The authors use DES methods by utilizing the Process Analyzer Function (PAN) in Arena to simulate Marines navigating through their training continuum from arrival to graduation (Davenport et al. 2007). The goal of the simulation was “to reduce that average waiting time experienced by Marines as they wait for their formal training to commence” (Davenport et al. 2007, page 1387). The queue that the Marines waited in is referred to as the Marines Awaiting Training (MAT) queue.

The authors ran 32 scenarios with solutions including a compacted training schedule, increasing the number of instructor billets, and varying the minimum or maximum allowable class sizes. The model implemented a 364-day waiting period to model the ongoing training system and ran through 1000 replications. The authors found that even under the worst-case scenario with basic changes being implemented, the MAT queue waiting time could be reduced by at least 37%, with the best-case scenario being 88% with all solutions implemented (Davenport et al. 2007).

The flow from arrival to graduation for the Marines is similar to the SWO for this thesis. The first key difference is the Marine arrival schedule, which is weekly and consistent. SWOs have seasonal accession rates that vary based on commission methods and cohort sizes. The model in this paper also has one specific wait queue, the MAT. There are two waiting queues in the model in this thesis, the wait for BDOC and the wait for JOOD. The maximum class size is also variable for MCCES while it is fixed for BDOC and JOOD courses. Statistics that this thesis include that are not included for MCCES are average number of students waiting as well as the number of man-hours spent waiting for training. More importantly, these statistics are offered at every stage of SWO training as opposed to overall training only.

3. Hourly Statistics Simio Model

This model is contained in the Simio support tab shown in Figure 2. This model was created by the Simio LLC and gathers statistics in an hourly time interval and tallies them so the simulation will record them in the results.

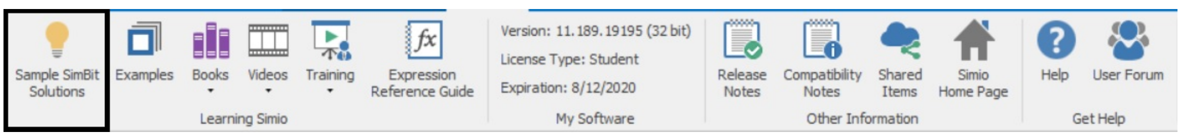


Figure 2. Simio support tab

The problem to be solved in this model is to know the average number of Entities in a Server's Input Buffer per hour of simulation. Figure 3 shows the model logic.

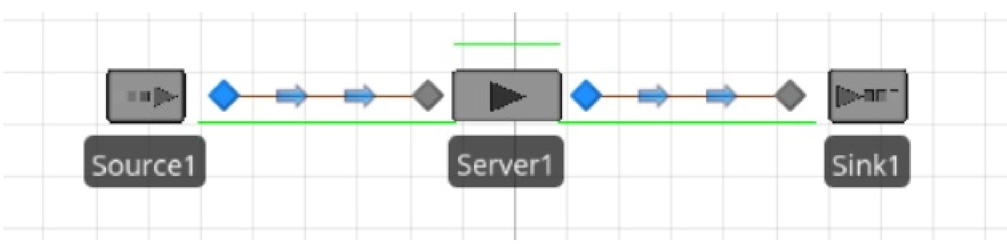


Figure 3. A common source-server-sink combination

The input buffer is the diamond node to the left of Server1. The approach is to create a timer that fires every hour. This triggers a process that contains a "Tally" step. This step keeps track of the AverageHourly statistic. This step is followed by an "Assign" step. This assign step reassigns two state variables whenever the process is triggered: AverageAtBeginning and TimeAtBeginning. These variables represent the average number of entities in the Input Buffer of the server, and the time at the beginning of the process trigger. Figure 4 shows the process.

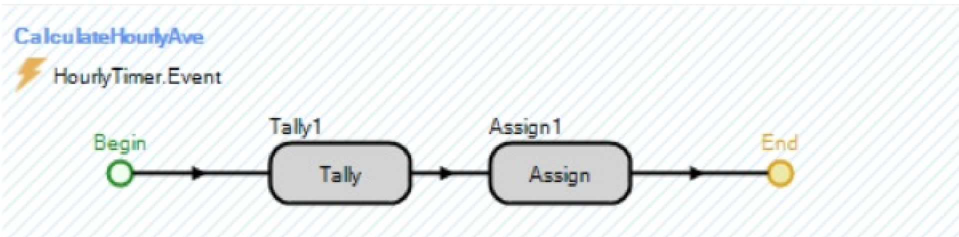


Figure 4. Tally and assign processes

The variables involved in calculating the AverageHourly statistic are as follows

- AveNumber—Average number of entities currently at the node
- CurrentTime—The current time of the simulation
- AverageAtBeginning—Average number of entities at end of the previous hour
- TimeAtBeginning—Time of the simulation at the end of the previous hour
- TimeInterval—The one-hour time interval

The tally statistic that represents the Hourly Average is then represented by the following equation:

$$\frac{\text{AveNumber} * \text{CurrentTime} - \text{AverageAtBeginning} * \text{TimeAtBeginning}}{\text{TimeInterval}}$$

Notice that the AverageAtBeginning and TimeAtBeginning are updated after the tally statistic is calculated in order to perform this calculation at the start of the process trigger. The results tab of Simio will now include this statistic.

The concept behind this process forms the basis for calculating monthly statistics in the Surface Warfare Officer model. A timer will trigger a statistical gathering process and gather these statistics in a monthly time interval. These statistics will represent average wait times, average number of students waiting at a node, and average man-days per month spent waiting. Obtaining statistics for class utilization rates will also be done in a similar manner.

C. PART III: MODEL FUNDAMENTALS

1. Fundamentals of Queueing Theory

Fundamentals of Queueing Theory discusses methods for mapping service demands (Gross et al. 2008). A very simple example is waiting in line for a service such as a bank teller or grocery store checkout. A topic that is relevant to the model in this thesis is series of queueing networks. The model in this thesis simulates a network of queues, or in this case a group of servers that represent training classes (BDOC and JOOD). Students enter the system at a source node, navigate from node to node in the system (these nodes are at the inputs or outputs of the servers), and depart the system at the sink node. This system, or network, is called a series queue. More specifically, the system that most closely mirrors the model in this thesis is a series queue with blocking. The nodes form a series system with the flow in a single direction. The units cannot proceed to station 2 without first visiting station 1. The capacity of these nodes, also called a waiting room, are assumed to be infinite. Figure 5 provides an example.

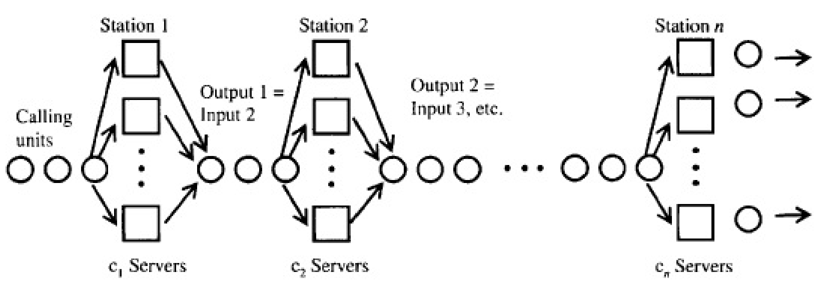


Figure 5. Series queue, infinite waiting room. Source: Gross, Shortle, Thompson, and Harris (2008).

Figure 5 shows different units in between each station, which could correspond to people or objects. These units go to different servers, which accept these units on a first come first serve basis. These servers could be a manufacturing process, a checkout counter, a cash register, etc. Which server the unit goes to depends on some probability. After going through the first station, the units line up and wait for the next station, where the next group of servers are. This continues until a stopping condition is met.

There are different ways to simulate a queueing network. The network must first be classified as open or closed. The model in this thesis is classified as open, since entities are created and introduced to the system, then leave the system after navigating through the servers. The simulation can also be a continuous or discrete simulation.

Each station in Figure 5 contains c number of servers. The calling units only need to visit one server before waiting in queue for the next station. Since this is a blocking queue, units will wait at the output node of station 1 until the previous calling units are done and station 2 opens for the next set of calling units. The system in Figure 5 is also known as a Jackson Network (Gross et al. 2008). It is a system that is formed by a finite string of queues where the entities in the system must visit each queue in order. With the Jacksonian Network, a path that the entity takes depends on probability rij, that is, the probability the entity is traveling from node i to node j. Key differences between the Jackson Tandem Network and the model in this thesis will be highlighted in Chapter III.

2. Basic Event Graph Modeling

This technical paper introduces the Event Graph methodology (Buss 2001). This methodology models event list logic. This logic directly corresponds to the logic of the SWO training pipeline. The simulation dynamics are driven by a list of future events, whereby the next event in the list gets executed. When an event executes, it updates state variables, if any, and schedules further events, if any. An Event Graph is a graphical representation of a model via events and their scheduling logic. An event, which corresponds to a node in the representation, may induce state variable transitions, and the directed edges connecting the nodes determine the scheduling of other events. Figure 6 demonstrates a basic Event Graph.

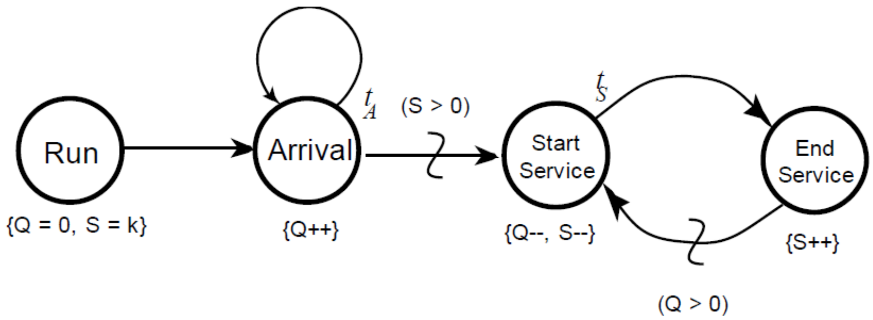


Figure 6. Simple event graph. Source: Buss (2001).

The figure shows four events, Run, Arrival, Start Service, and End Service. The figure also shows the time distribution between arrivals, t_A and the time to complete service, t_s. The algorithm will execute the next event in line. The first event, Run, begins the simulation and sets the initial number of people waiting in queue, Q, equal to zero and the number of available servers, S, equal to some constant k. The event Run begins the simulation. It then schedules the Arrival event with a delay t_A from the current simulation time, which represents a person arriving to the system, therefore raising the number of people waiting in queue, when the Arrival event executes. The Arrival event then self-schedules another Arrival event. This ensures that objects continue entering the system. When the Arrival event executes, if the number of available servers is greater than zero, the Start Service event is scheduled with no delay. When the Start Service executes it lowers the number of available servers and number of people waiting in queue, and schedules the End Service event with delay t_S units of time. When the End Service event executes it raises the number of available servers, since the service is now complete. Also, if there are people waiting in queue, a Start Service event gets scheduled with no delay. Figure 7 provides an illustration of how Python executes the Arrival process.

```
def Arrival(self):
        new_arrivals = 1
        self.number_in_queue += new_arrivals

        self.schedule('Arrival', self.interarrival_time_generator.generate())

        if self.number_of_servers > 0:
            self.schedule('Start_Service', 0.0)
```

Figure 7. Arrival process in DESpy

The event graph of Figure 6 sets the foundation for the event logic that the Python model follows. The arrivals will follow the historic commission rate from the projected 2019 SWO accessions. The services will correspond to the BDOC and JOOD classes that the SWOs attend. The Event Graph corresponding to the DIVO training pipeline will be discussed further in Chapter III of the report.

D. SUMMARY

Part I of this chapter discussed two overarching themes behind the necessity in implementing another course prior to SWOs going to the fleet. The USFF Comprehensive Review outlined how underprepared the individuals involved in the 2017 mishaps were and the general view of how sailors struggled to handle emergency situations. The U.S. Readiness Report discussed the Navy's "must-do" culture. While this culture leads to hardworking sailors ready to handle substantial workloads, it relies on the fleet to handle extra certifications and qualifications, overstressing the fleet and lowering its readiness. Part II discussed work related to the concepts used in this thesis. These works included DES projects for improving training pipelines for ADF manpower and the U.S. Marine Corps communications-electronics school. Part II also discussed a Simio model that uses a similar statistical gathering process utilized in the Simio model of this thesis. Part III provided logical concepts that are instrumental to the model in this thesis. These concepts included queueing theory and the event graph logic used in Python. Chapter III puts these technical concepts together to describe the SWO model.

THIS PAGE INTENTIONALLY LEFT BLANK

III. METHODOLOGY

This chapter discusses all the programs used to accomplish the model of this thesis and centers around the overview of the program along with the features that will be used in the analysis. The Simio program produced the results and analysis outlined in the following chapter. This chapter also outlines the assumptions that go into the model: entity generation and travel logic from beginning to end. The connection between the simulation program and Excel is introduced to outline how the model imports data. The final section of this chapter displays the event graph for the Python model followed by a description of the logic.

A. INTRODUCTION

Figure 8 shows the breakdown of JOOD wait times for 2019.

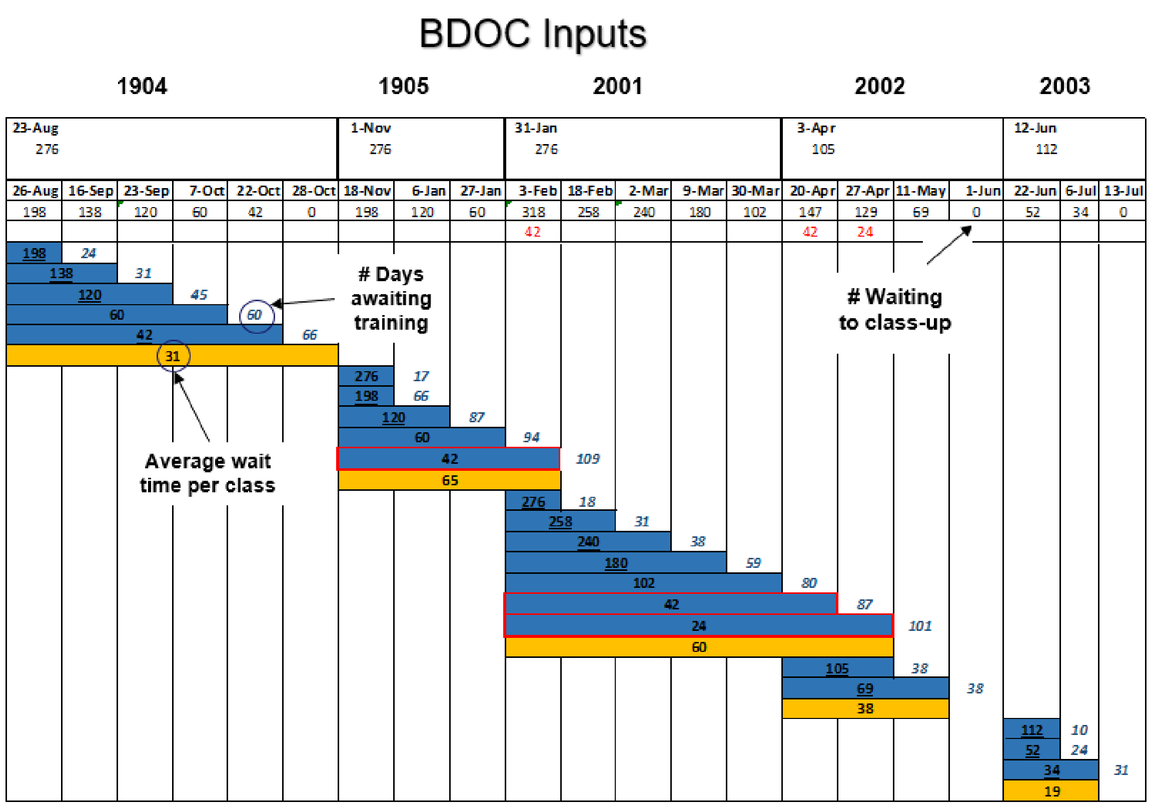


Figure 8. JOOD wait statistics

There was a total of a little over 1,000 accessions for the SWO community is 2019. Days waiting for training and the number of students waiting to class up varied widely over the year. Number of students waiting to class up for JOOD ranged from 0 students to over 300 students throughout the year. Days awaiting JOOD training ranged from 10 to 109 days on average. No data has been provided for BDOC waiting periods.

The model implements data provided by PERS-41 in order to analyze monthly metrics. These metrics include number of students waiting and average time waiting at various points in training, class utilization rates for BDOC and JOOD, and confidence intervals. Direct consequences to any proposed changes to the schedule, classroom capacities and sizes, or any other additions can be observed with ease when run in the model.

B. THE SIMIO PROGRAM

Simio is an entity based simulation program. This software allows building models by combining "objects." Objects represent physical elements of the overall system. In this case, the elements are the training courses that SWO officers are required to take and the officers themselves. The following list defines nomenclature used continually throughout this report:

- Server—An object that represents a service station (classroom trainings BDOC and JOOD)
- Entity—A movable unit that travels through the system (students)
- Source—Object that creates the entities (commissioning source)
- Sink—Object that destroys the entities (students entering the fleet)
- Connector—Zero-time travel link between nodes
- Process—Series of steps that determine object behavior at specific points in the simulation (Pegden and Sturrock 2013)

Another thesis that utilized Simio as the preferred simulation software outlined the following reasons for its use:

- Allow concurrently running interactions and processes with a large number of entities and objects
- Define and change attributes for entities, objects, and global variables
- Use entity variables, global variables, and mathematical expressions in the decision logic
- Import and export data from other applications (e.g., an Excel Spreadsheet)
- Offer concurrent animation that displays key elements of the system dynamically travelling through the system in real time, to help assist in debugging and model validation (Gable 2014)

In addition to the aforementioned attributes, the following attributes are also required to:

- Generate entities whose number and time of generation follow a specific schedule
- Allow these entities to interact with servers that also follow a strict calendar schedule throughout the year while being able to run the simulation within a specific time interval

All outputs generated from single runs and experiments from Simio were imported in the statistical package, R, for analysis. The output from Simio are in the form of pivot tables in a comma separated value (csv) format.

1. Model Logic

Figure 9 outlines the logic used in designing model in Simio. All the JOOD courses are enclosed in a dashed box. This box indicates that JOOD is what is different from the original training of the SWO prior to their first DIVO tour. Figure 10 outlines how the entity generation rate for the model. Commission rates are determined based on historic rates.

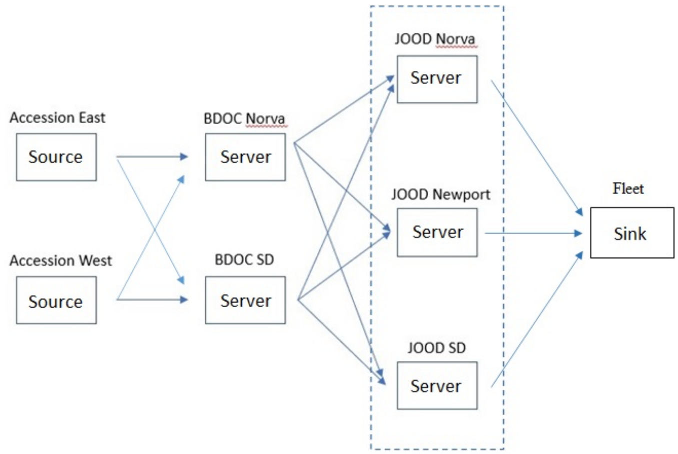


Figure 9. Travel logic for model entities

Figure 10. Monthly commissions based on 2019 data

The numbers at the top represent the summation of commissions in each month. PERS-41 provided a spreadsheet that provides more detail on what date the commissions arrive during each month. The spreadsheet also indicates if the commission is an East or West entity. Eastern commissions attend BDOC in Norfolk while Western commissions attend BDOC in San Diego. Although if, for example, BDOC in one location is filled with officers while the other BDOC has vacancies, the model will allow officers from one coast to attend the other BDOC in order to minimize wait time. JOOD classes are in Newport, Norfolk, and San Diego. Upon completion of BDOC, students can attend any of these courses, depending on what opens first, and which students have been waiting the longest time. Once a student commissions or completes BDOC training, the student will wait at the output of that training until classing up to the next training. To introduce randomness to the model, the number of arrivals input to the model takes the form of a uniform distribution. The uniform distribution is justified in the section that discusses the assumptions of the model. The upper and lower limits of the random uniform distribution allow for an approximate 10% standard deviation to allow enough randomness to the model to generate confidence intervals for all the output statistics. Every other input is provided by PERS 41. The logic of Figure 9 can also be updated to incorporate any section of the SWO training pipeline that is desired to analyze.

2. The Model

Figure 11 and Figure 12 display the two primary pieces to this program that run the logic outlined in Figure 9.

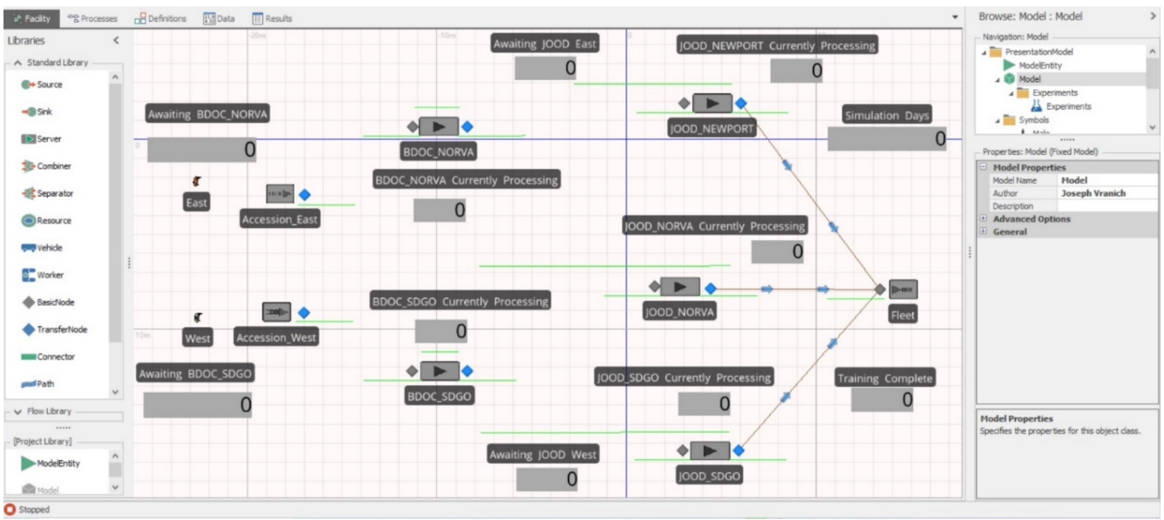


Figure 11. Overall model

The sources at the beginning of this model create the East and West entities. Upon creation, the entity travels to the output node of the source where they wait for their respective BDOC training to begin. Since there is no connector between the source and the BDOC server, the entities are transferred via a process. This process is considered a "wait-transfer" process, where the entity waits until the simulation has reached the start date of a training, then transfer to the input buffer of the server. The server capacity is updated on that date to take in the same number of students as the classroom size allows. Upon completion of the BDOC server, the entity travels to the output buffer and go through the same process to be transferred to the JOOD server. Upon completion of the JOOD training, the entity transfers to the output buffer as before. The entity then instantly travels to the sink via a connector. At this point, the entity is destroyed and gone from the system, representing the conclusion of their training.

Figure 12 shows the start date of these classes that allow the transfer of these entities.

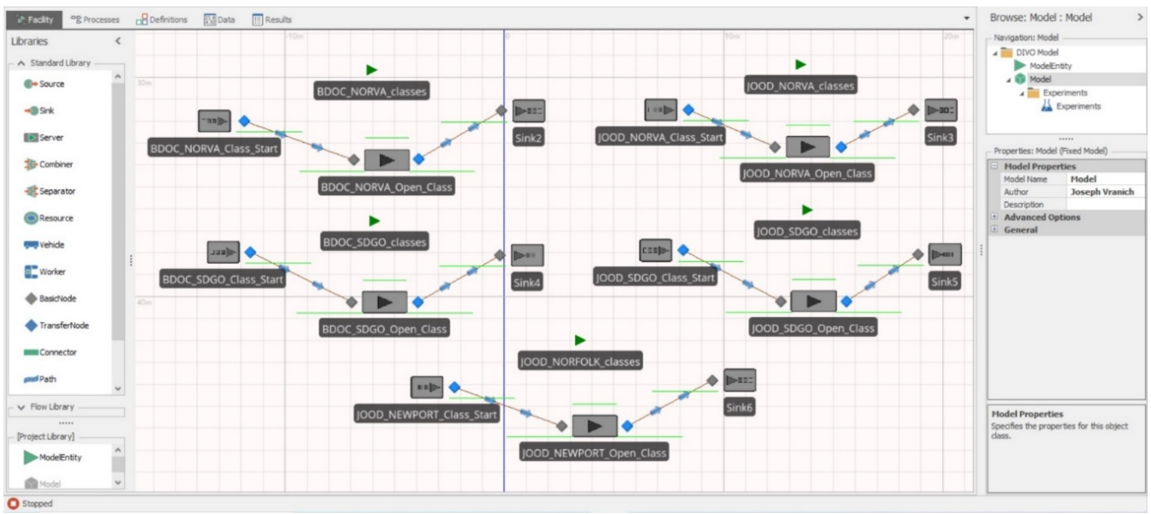


Figure 12. Training counters

Figure 12 shows five source-server-sink combinations. These five combinations represent the five classes (two for BDOC and three for JOOD). A class entity is generated by the source on the start date for that class. This event allows for the transfer of East and West entities from one training to the next. It also begins a counter that keeps track of which class has begun. This counter is necessary because several classes have different durations depending on the time of the year. The entity entering the server triggers a process that uses a “Fire” step in order to start an event. This event is the opening of the class that is represented by the server. When this event fires, another process that contains a “Wait” and “Transfer” step combination begins. Figure 13 and Figure 14 outline these processes.

Figure 13. Fire process

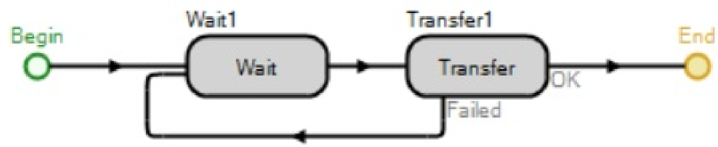


Figure 14. Wait-transfer process

Figure 13 is a process that consists of only one “Fire” step, which begins once an entity is generated on a class convening date and travels through a class opening server in Figure 12. The “Fire” step triggers the “Wait-transfer” process in Figure 14. If there are entities that the “Transfer” step fails to transport, then the process follows the Failed line and feeds back into the “Wait” step. For example, if a BDOC class in San Diego opens, the “Wait” step will proceed to the “Transfer” step. This will transport the entities from the Output Buffer of the Accession West Source to the Input Buffer of BDOC San Diego Server. Only the number of entities that can be handled by the server capacity will be transferred, while the rest will remain in the source input buffer. This feature is necessary in order to calculate the average wait times and number of students at this phase of their training.

The logic to transfer entities from BDOC to JOOD is more complex. There are three locations that entities can travel to upon completion of BDOC. The goal is to send students that have been waiting the longest time to one of these trainings first. That is accomplished by having the program look for which BDOC output node (where the entities wait upon completion of BDOC) contains the largest number of entities, which is typically the case when entities have been waiting longer. This will cause the program to appropriately alternate between BDOC waiting rooms when gathering entities for transfer to JOOD. Figure 15 shows the logic of this process.

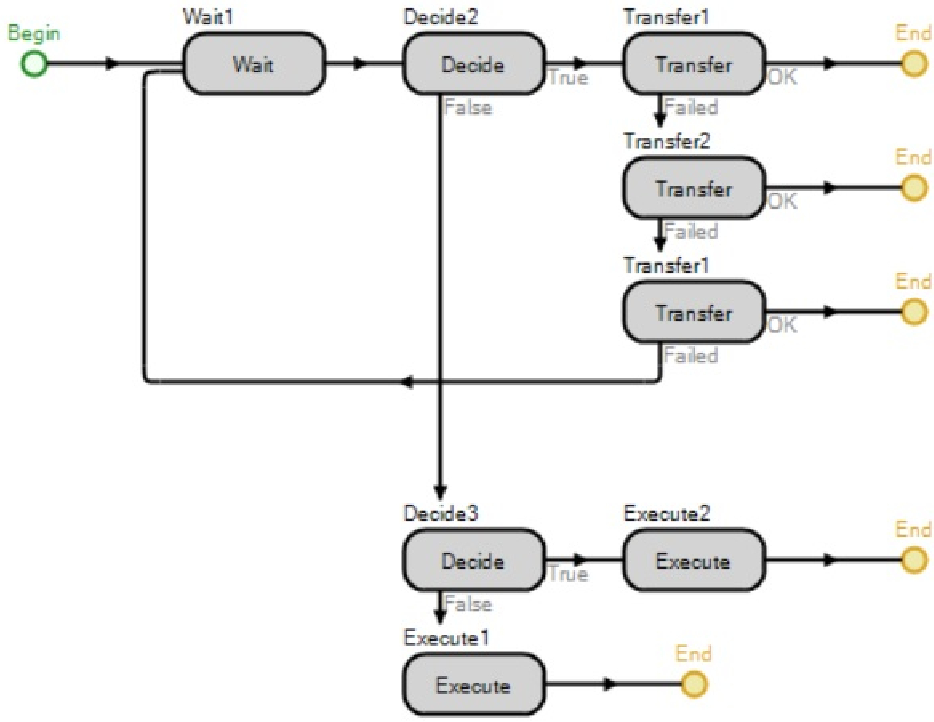


Figure 15. Norfolk BDOC to JOOD transfer logic

The process begins with the same "Wait" step as Figure 14. This step feeds into a "Decide" step. The "Decide" step is verifying whether a condition is true. In this case, it is verifying if the number of entities waiting in Norfolk is larger than in San Diego. If true, it goes through a parallel combination of "Transfer" steps in order to find which JOOD class it is meant to travel to, and the program gathers the entities waiting at that location. If false, the path feeds into another "Decide" step directly below, which verifies if the number of people waiting in Norfolk is less than all the open seats available for the opening JOOD classes. If true, the logic performs an "Execute" step that triggers another process that looks identical to Figure 15. The process that is triggered is the transfer logic for travel from San Diego BDOC to JOOD. If the condition is false, it goes to another "Execute" step below, indicating that the number of people waiting after Norfolk BDOC is greater than the number of available seats for JOOD, which triggers an "Execute" step that begins the logic in Figure 16. The purpose of the process in Figure 16 is to transfer the entities to any JOOD class that is available, since there are not enough JOOD seats to accommodate all students waiting.

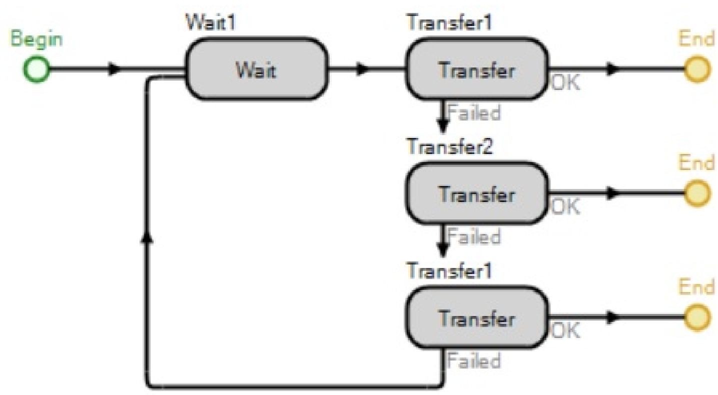
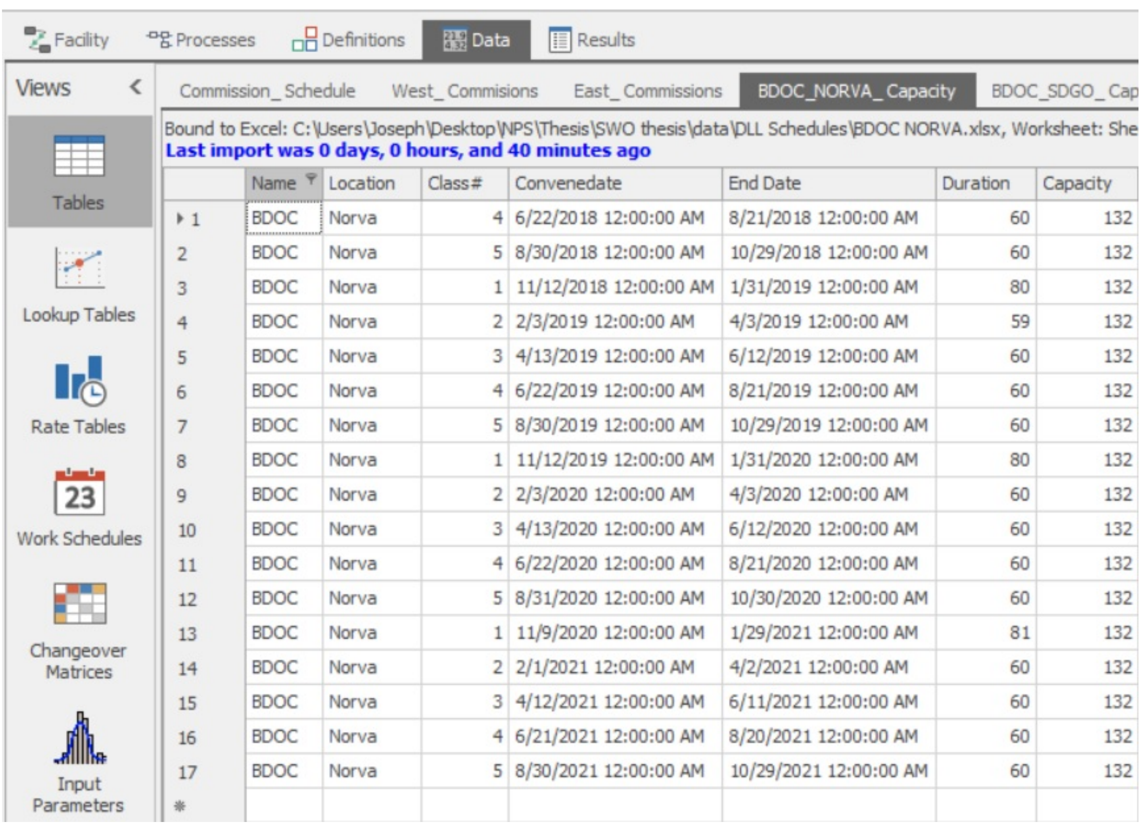


Figure 16. General BDOC to JOOD transfer process

The firing of open class events that trigger the transfer processes are based on schedules for BDOC and JOOD provided by PERS-41. Figure 17 shows an example of a BDOC schedule.



	Name	Location	Class#	Convenedate	End Date	Duration	Capacity
1	BDOC	Norva	4	6/22/2018 12:00:00 AM	8/21/2018 12:00:00 AM	60	132
2	BDOC	Norva	5	8/30/2018 12:00:00 AM	10/29/2018 12:00:00 AM	60	132
3	BDOC	Norva	1	11/12/2018 12:00:00 AM	1/31/2019 12:00:00 AM	80	132
4	BDOC	Norva	2	2/3/2019 12:00:00 AM	4/3/2019 12:00:00 AM	59	132
5	BDOC	Norva	3	4/13/2019 12:00:00 AM	6/12/2019 12:00:00 AM	60	132
6	BDOC	Norva	4	6/22/2019 12:00:00 AM	8/21/2019 12:00:00 AM	60	132
7	BDOC	Norva	5	8/30/2019 12:00:00 AM	10/29/2019 12:00:00 AM	60	132
8	BDOC	Norva	1	11/12/2019 12:00:00 AM	1/31/2020 12:00:00 AM	80	132
9	BDOC	Norva	2	2/3/2020 12:00:00 AM	4/3/2020 12:00:00 AM	60	132
10	BDOC	Norva	3	4/13/2020 12:00:00 AM	6/12/2020 12:00:00 AM	60	132
11	BDOC	Norva	4	6/22/2020 12:00:00 AM	8/21/2020 12:00:00 AM	60	132
12	BDOC	Norva	5	8/31/2020 12:00:00 AM	10/30/2020 12:00:00 AM	60	132
13	BDOC	Norva	1	11/9/2020 12:00:00 AM	1/29/2021 12:00:00 AM	81	132
14	BDOC	Norva	2	2/1/2021 12:00:00 AM	4/2/2021 12:00:00 AM	60	132
15	BDOC	Norva	3	4/12/2021 12:00:00 AM	6/11/2021 12:00:00 AM	60	132
16	BDOC	Norva	4	6/21/2021 12:00:00 AM	8/20/2021 12:00:00 AM	60	132
17	BDOC	Norva	5	8/30/2021 12:00:00 AM	10/29/2021 12:00:00 AM	60	132

Figure 17. Typical class schedule

The table in Figure 17 was created by an Excel file linked to the Simio table tools tab. Notice that durations are slightly different depending on the class start date.

In order to calculate the monthly statistics for this model, the model uses the logic of the Compute Hourly Statistic Model. A monthly timer is created in order to be the firing mechanism for the statistic gathering process. This process included twelve “Decide” steps that each contained Boolean statements. These Boolean statements check which month the simulation is in at the time of the process trigger. This process is shown in Figure 18.

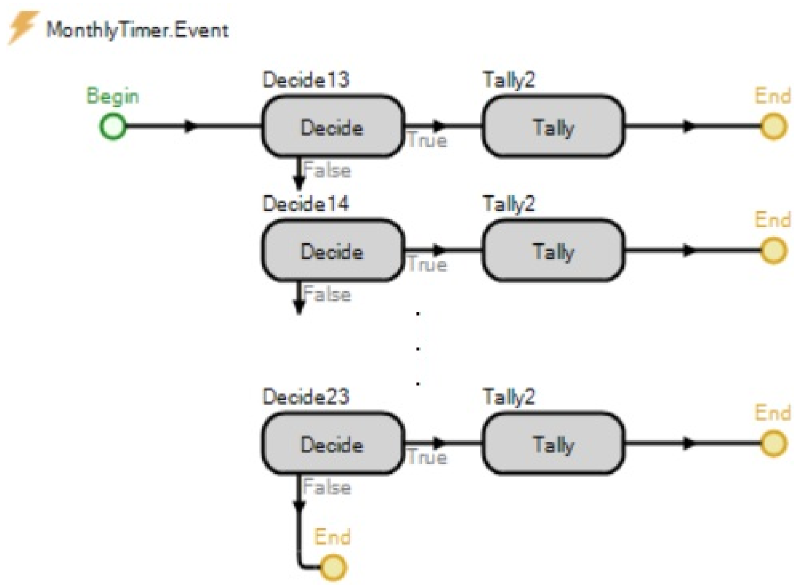


Figure 18. Tally process

The MonthlyTimer.Event in the top left corner indicates that a monthly timer is what triggers this process. If the month in the simulation is different than what a “Decide” step is checking for, than it is false and will check the next “Decide” step which is verifying a different month. Depending on the month, the program would proceed to calculate statistics with a “Tally” step for that time period, which could be collected in the resulting Simio pivot tables. The process in Figure 18 illustrates how the model calculates average students waiting and average time waiting per student per month and the class utilization rate per class.

a. Inputs

The inputs to the model are the number of commissions for Eastern and Western students, the date of commission, starting dates and capacities for BDOC and JOOD courses. All these inputs are recorded on multiple Excel spreadsheets. These spreadsheets can be imported into Simio via a Direct-Link Library (DLL). A DLL allows a table to attach itself to a spreadsheet file (xlsx, csv, etc.) and take on its properties. Once the link is established, the spreadsheet can be imported to populate a Simio table. These tables contain the class schedules, durations for each class, classroom capacities, and commissioning rates throughout the year. Any changes can be made in these spreadsheets and reuploaded to the program quickly to allow non-analysts to run simulations and observe the results of these changes.

b. Assumptions

The average amount of commissions each year is about the same as the commissions for 2019, although slightly different due to randomness. The model assumes that these rates remain the same from year to year to achieve steady state conditions. The number of commissions on a date vary based on a random uniform distribution, where the upper and lower limits are 10% off from the mean. The uniform distribution is justified because the upper and lower limits of the number of officers that are expected to commission are equally likely for each date. For example, May 25th is the typical date where USNA graduates commission. In 2019 there were 250 commissions. It is equally likely that, in a given year, there can be 240 graduates. Following this logic, the model assumes that the number of commissions for a given date are equally likely between 10% above and below the mean. The dates of commission and the dates for BDOC and JOOD training remain the same each year.

The actual time of travel from station to station is instantaneous. Because of this, there is no consequence to an entity travelling from Norfolk to San Diego versus remaining in Norfolk for multiple trainings. Depending on the type of job they perform, certain SWOs also attend BST in their initial training phase. This training is considered negligible for the model, however, as it only applies to certain SWOS and therefore is not included in the logic.

C. THE PYTHON PROGRAM

Python offers many of the same attributes that Simio offers. The primary difference is the visualization aspect. Simio provides animations of desired statistics throughout the simulation with ease. Visualization within the Python program is more difficult. Python, however, offers the benefit of having more control. As the processes and restrictions become more complex, this control is needed. The goal of creating the DIVO model in Python is to provide a proof of concept model in DESpy. The results of the DESpy model are compared to the Simio model in Chapter IV to show that the model works. As mentioned in Chapter II, the ultimate aim of this project is to model other aspects of the SWO pipeline, to include DH training. The discussion of DH training as well as improvements to the existing model is included in Chapter V.

D. EVENT GRAPH

The logic for the simulation is illustrated in Figure 19. The logic for the SWO simulation follows the same format as Figure 9. However, the schedule for the BDOC and JOOD trainings are set and unchanging, therefore there are no conditions to start a particular class other than time considerations. The inputs and assumptions for the Python model are the same as the Simio model. There are some minor logical differences that are due to the Python model being hardcoded and the Simio model being built with process blocks that may be unaccounted for.

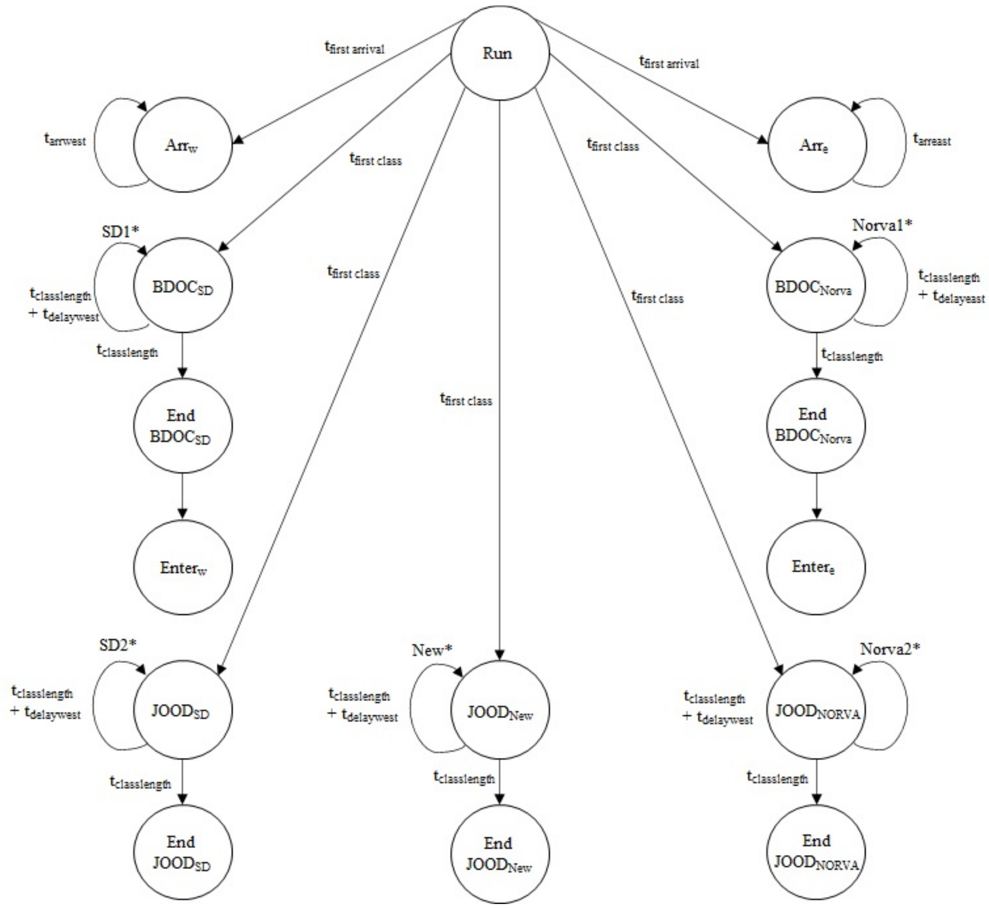


Figure 19. Model logic event graph

The model begins with the *Run* event. The Run event is the starting event that schedules seven other events. These events are the Western and Eastern arrivals, BDOC in San Diego and Norfolk, and JOOD in San Diego, Norfolk, and Newport. The delay time to the first of these events is all known based on historical data. Once these events begin, they loop back on themselves a certain time later. For the *Arr* events, the delay is the time to the next arrival. For BDOC and JOOD, the time to retrigger the event is the length of time the class is in session plus the time in between that class and the next class. After a class length has gone by, the End BDOC or End JOOD event begins. The reason these events exist, as well as the Enter events that occur after the End BDOC class, is because these events gather statistics for the simulation. These statistics include number waiting in queue, time waiting in queue, and the number of people that complete the training in a given period of time. The asterisks in the model indicate that there is a particular logic that each class follows. The pseudocode for this logic is listed in the appendix.

The logic of this model largely follows the same logic as the Simio model. Western arrivals will attend BDOC in San Diego, while Eastern arrivals will attend BDOC in Norfolk. However, if there are openings for BDOC on an opposite coast while students are waiting, the entities can go to a BDOC class on the opposite coast. Students that complete BDOC can attend any JOOD class depending on what opens for and how long they have been waiting to minimize wait time. The logic that determines where entities travel is in the appendix.

E. SUMMARY

The overall model outlined in this chapter is meant to simulate the DIVO training path of a typical SWO. Once the SWO commissions and is classified to either an Eastern or Western port, they travel to BDOC and then to JOOD based on classroom availability and priority. All the data used as inputs to the model are provided by PERS-41. Chapter IV presents the results of the model logic outlined in this chapter through monthly statistics and class utilization rates. The results are taken from the Simio simulation. Average overall wait times and queue sizes are compared to the Python model as well.

IV. RESULTS

This chapter discusses the results from a single simulation run and the stochastic analysis following multiple experiments in Simio. The analysis and the monthly statistics calculated come from the data provided by PERS 41 and estimating the commission rates based on historical data from a one-year time period. The single simulation run is performed in order to observe the graphical behavior of student behavior in real simulation time. This provides indication of what time periods experiences the most friction. Multiple experiments are also run in order to account for years with high and low accession rates and provide a 95% confidence intervals (CI) for each measure of performance. The findings will be presented in terms of the number of students waiting, the wait time of the students, and the man-days per month spent waiting. These will be provided on a monthly basis starting in May, as that is when the most accessions happen every year. Class utilization rates are also explored. These results all lend to insights on how to alter the training schedule in order to minimize friction.

A. SINGLE SIMULATION RUN

Figure 20 shows status graphs that illustrate the cyclic nature of the number of students waiting.

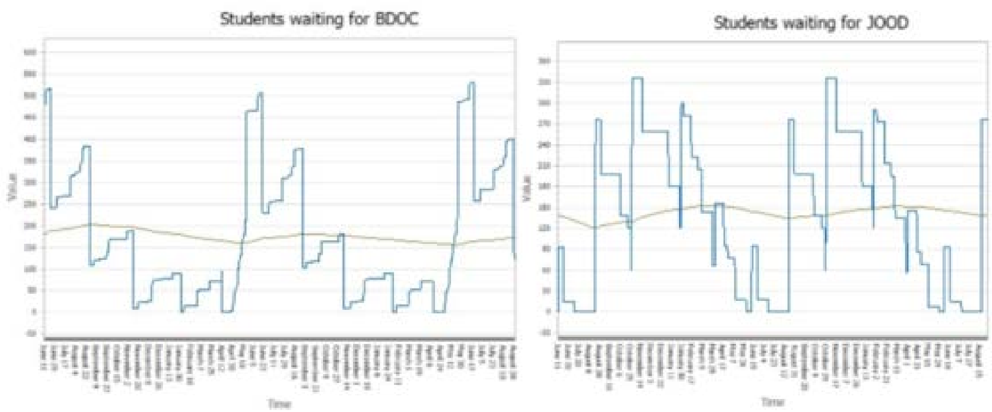


Figure 20. Number of students waiting for BDOC and JOOD in one simulation run

The blue line represents the number of students and the green line represents a running average, starting from the beginning of the simulation. The simulation was run for three years with statistics being gathered after 150 days in order to allow for a warm-up period. This period let students enter the system before achieving steady state sometime later. Steady state is not present at the beginning because the simulation starts with zero students in the system. It takes about six simulation months for students to begin or complete JOOD training from the beginning. Steady state in this context means that students are entering and leaving the system in a stable manner based on commission rates and training schedules being constant every year. The plots show the absolute number and a running average. The plots show that students waiting for BDOC and JOOD are both a stable cycle. That is, once steady state is reached, the number of students waiting for training will reset. In an unstable queue, the number of students would continue to rise every year, eventually leading to infinite wait times in the long run.

BDOC has a large backlog of students waiting between May and August. JOOD experiences many students waiting between August and March. There are three large drops in the number of students waiting for BDOC. These drops occur in June, August, and November. This is due to class starting. The conclusion of these classes in August, October, and January, correspond to the largest jumps in the number of students waiting for JOOD training.

B. REPLICATIONS

The model provides results based on 50 independent replications, each lasting an average of 0.35 seconds on a Samsung with an Intel Core i7-8550U CPU and 16.0 GB installed RAM. The overall schedule of commissions for Eastern and Western accessions are unchanging. The commissioning dates do not change because they are based on various commissioning sources such as USNA or NSTC which have rigid schedules. The primary change is the number of commissions throughout the year. Confidence intervals are calculated assuming a t-distribution with a 95% CI and n – 1 degrees of freedom. The n indicates the number of experiments run for the results. Each replication incorporates a 150-day warmup period to account for the number of entities in the system at the beginning of the simulation being zero. Several months are required for students to complete training in order to observe steady state conditions. The conditions are observed to be in steady state when the pattern of students entering and leaving the system are unchanging, as observed in Figure 20.

1. Number of Students Waiting, Wait Times, and Man-Days

The primary monthly statistics of interest were the average number of students waiting for BDOC or JOOD each month, average time waiting for BDOC or JOOD for each student each month, and the number of man-days spent waiting for each month. The man-days spent waiting are calculated by multiplying the number of students waiting by the time waiting per student in days. This statistic is calculated within the simulation in order to incorporate any dependencies experienced between these variables. Figure 21 shows the plots of these statistics and 95% confidence intervals.

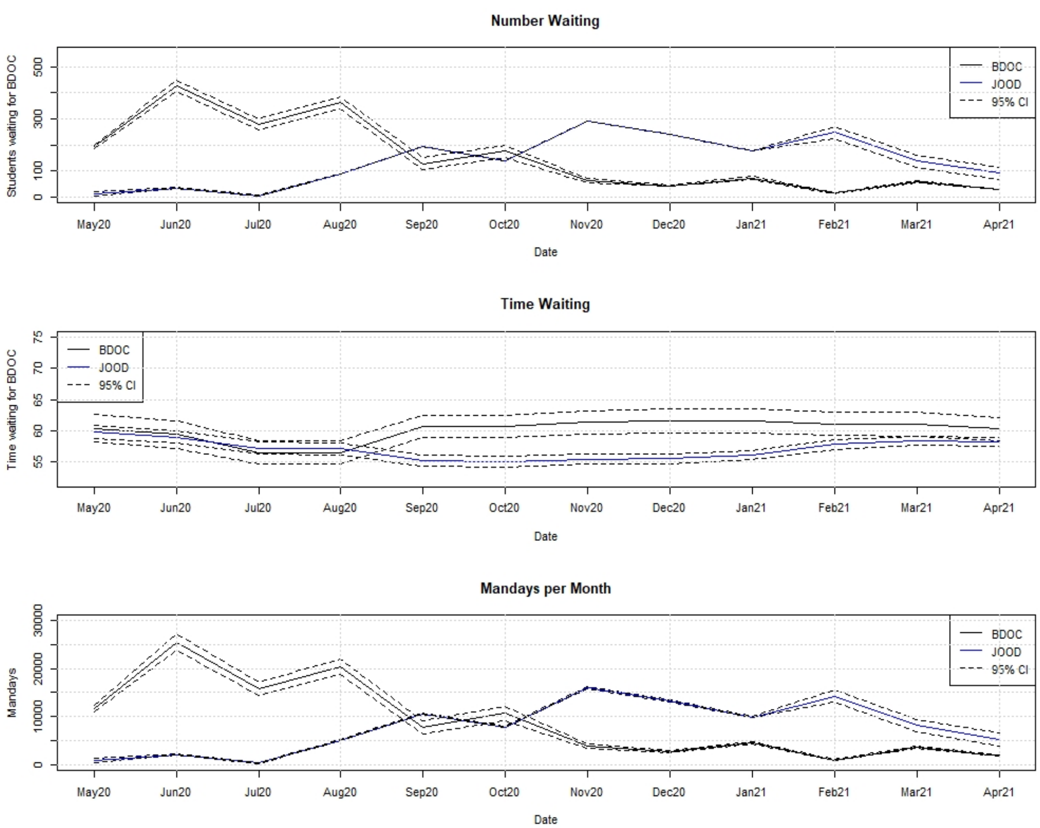


Figure 21. Primary monthly statistics with 95% CI

The first thing to notice is that the largest number of students are waiting for BDOC during the summer months. This is because the largest number of officer commissions occur in May, followed by the second largest number of officers commissioning in June. This leads to a build-up of students awaiting their first training. Since commission rates drop throughout the year and classroom capacities are large for BDOC, the number drops rapidly from month to month throughout the year. The opposite is true for JOOD. As the summer BDOC classes conclude, this leads to many students waiting for JOOD, therefore these numbers rise, before dropping again in the month of May. This is consistent with the results found from the single run real-time plots in Figure 20. The average amount of time waiting for BDOC and JOOD is relatively constant throughout the year.

The plots also show that the confidence interval for BDOC is much wider than the confidence interval for JOOD. This is due to the variability in student commissions each month, which leads to more uncertainty. With JOOD training however, the number of students and time waiting is much more certain due to being confined within a strict schedule. For example, the highest commissioning rate occurs in May. Therefore, the next couple of BDOC classes are full in every simulation. As a result, the number of students waiting for JOOD are the same for the following months. This why the standard deviation for students waiting for JOOD is zero from August to January. The uncertainty for BDOC and JOOD numbers and wait times can be summed up through their relative errors and summarized in Table 1.

Table 1. Relative errors per month

BDOC			JOOD		
Month	Number Waiting	Time Waiting	Month	Number Waiting	Time Waiting
May	0.032	0.037	May	0.644	0.017
Jun	0.049	0.036	Jun	0.070	0.016
Jul	0.077	0.033	Jul	0.360	0.016
Aug	0.062	0.033	Aug	0.000	0.017
Sep	0.178	0.029	Sep	0.000	0.016
Oct	0.130	0.029	Oct	0.000	0.016
Nov	0.112	0.030	Nov	0.000	0.015
Dec	0.073	0.031	Dec	0.000	0.014
Jan	0.066	0.031	Jan	0.008	0.014
Feb	0.073	0.031	Feb	0.093	0.013
Mar	0.057	0.031	Mar	0.166	0.012
Apr	0.047	0.030	Apr	0.261	0.012

The relative error is in decimal form. Multiplying each row by 100 would give the percentage.

The relative error represents the measure of uncertainty of a statistic compared to its actual size. The purpose of the measure is to put perspective on the error. For example, an error of one day would be large if the total length was two days and small if the total length was one month. The relative error is calculated by taking the halfwidth of the CI and dividing by the center. The maximum relative error for the number of students waiting and time waiting for BDOC is in September and May, respectively. The maximum relative error for JOOD occurs in May for the number of students waiting and in May and August for time waiting. Although, the relative errors for the number of students waiting in May and June for JOOD are somewhat deceptive, given the number of students waiting on average for JOOD in these months are very close to zero, which Figure 20 illustrates.

Table 2 shows the wait time summary statistics for East and West Coast entities.

Table 2. Wait time statistics (days)

	East BDOC	West BDOC	East JOOD	West JOOD	East Overall System	West Overall System
Average	51.477	68.254	54.569	57.802	191.779	216.713
Std.Dev	1.187	0.869	0.200	0.796	1.425	1.677

West overall experiences longer wait times and overall time in system due to the Simio program choosing to send these entities to JOOD first. Simio sends them first because there are more West commissions overall, so they tend to have priority.

Table 2 illustrates the wait times, in days, for Eastern and Western entities at each significant point in the system. The table illustrates an average wait time 191.78 ± 2.79 days for Eastern accessions and 216.71 ± 3.29 days for Western accessions based on 50 degrees of freedom and a 95% confidence interval. The numbers show that the Western accessions have higher mean time wait times at each point. This is due to the higher accession rate for Western entities. As a result, more entities tend to be waiting for training, driving up the average wait time. This also leads to Western entities spending more time in the system overall compared to Eastern entities. While this seems like a long time, the overall time spent in training makes up a significant fraction of their time in the system. Training for BDOC is 60 days, while training in JOOD tends to be about 25 days, depending on the time of year. For example, training is equivalent to 32 days when it straddles Thanksgiving. This model also does not consider Billet Specific Training, which would lengthen the amount of time the entities spend in training and in the system overall. Table 3 summarizes the fraction of time each type of entity spends waiting with a 95% CI.

Table 3. Fraction of time waiting for training

	Lower Bound	Average	Upper Bound
East	0.547	0.553	0.559
West	0.577	0.582	0.587

The upper and lower bound for East and West were calculated using a t-statistic with 49 degrees of freedom and a 0.05 confidence level.

The table shows that for East and West commissions, more time is spent waiting than time spent in training.

2. Class Utilization Rate

Another metric of interest is class utilization rate. Class utilization rate is defined as the proportion of class capacity used for each class throughout the year. Looking at the status plot for students waiting for BDOC in Figure 21, the sudden drop corresponds to students being admitted to class. The final two drops are shorter, indicating the classrooms are not being filled up all the way. The following plots show the percent utilization by class for BDOC and JOOD.

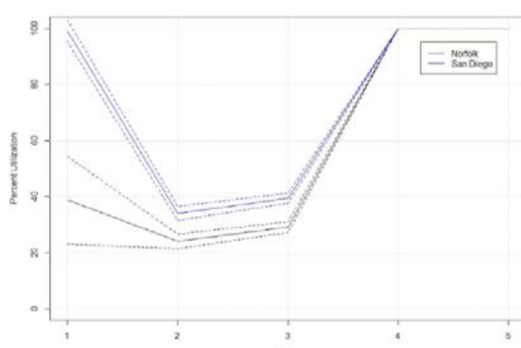


Figure 22. Percent utilization per BDOC class

The plot shows that classes 2 and 3 have low utilization rate for both Norfolk and San Diego, less than 50%. Class 1 for Norfolk is also low, but with a much wider confidence interval. The BDOC classrooms in San Diego tend to fill up for class 1. The students that are leftover are sent to the BDOC class in Norfolk. If there are a lot of commissions each year, the number of leftover students waiting for San Diego classes will be higher, therefore Norfolk will have higher utilization rates as well. If there are few commissions, the utilization rates will be lower in Norfolk. This is what leads to such a wide interval for class 1 for Norfolk.

San Diego tends to have a higher rate of class utilization compared to Norfolk for every class that does not have 100% utilization. This is due to the rate of Western commissions. Of all the accessions that happen every year, approximately 60% of those accessions are slated for Western homeports. These accessions attend BDOC in San Diego. As a result, BDOC in San Diego tends to have a higher classroom utilization rate throughout the year. Class 4 and 5 happen in June and August respectively. This is following the high commission rates of May and June. Therefore, the utilization rates are always 100% for these classes. Both Figure 23 and Figure 24 show the same analysis for JOOD.

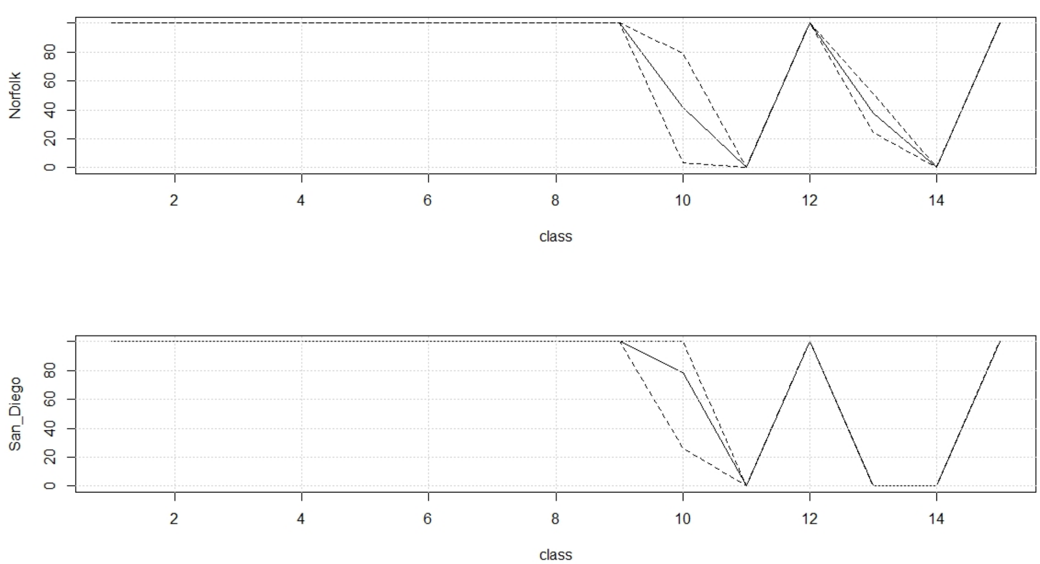


Figure 23. Percent utilization per JOOD class (Norfolk and San Diego)

Norfolk and San Diego are stacked on top of each other because they contain the same number of classes per year, while Newport has only 11. Therefore, Newport requires its own graph.

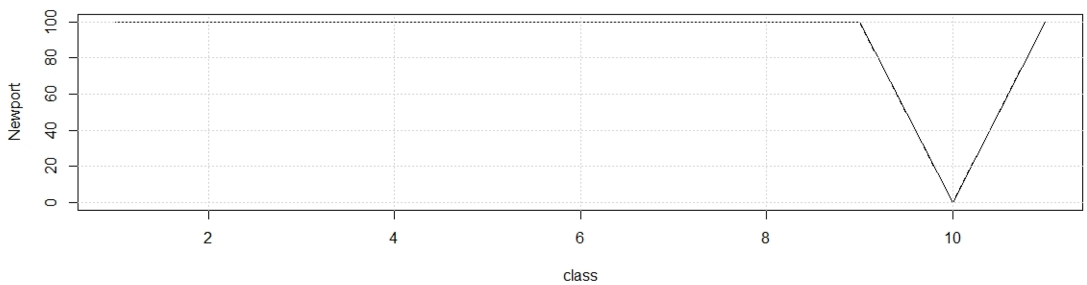


Figure 24. Percent utilization per Newport JOOD class

Figure 23 and Figure 24 show that the percent utilization of the majority of JOOD classes is 100%. However, utilization drops as the classes progress toward the end. Class 10 for San Diego and Norfolk have a large degree of uncertainty associated with them, with class 11 always being empty. Class 14, as well as class 10 for Newport, are also empty every time. This corresponds to August 3rd. This is consistent with Figure 20, which shows that the number of students waiting for JOOD tend to be low, or even zero, between the months of May and August.

Recall, the logic of the model is to send the entities to whichever JOOD class opens first, regardless of geographic location. As a result, these graphs show the highest possible class utilization rates. These plots show that if the class schedules remain the same very year and the number of accessions remain constant every year, there will be at least six classes that are empty throughout the year. This indicates that the schedule could be more efficient, or that the SWO community could take on more accessions each year to utilize the amount of space available. Discussion of potential schedule changes is provided in the sensitivity analysis.

3. Sensitivity Analysis

According to Table 1, most of the time each entity in the system is spent waiting for training. Since the amount of time spent in the actual training is beyond our control, emphasis will be placed on lowering the overall amount of time spent in the system when exploring schedule changes. Since this is the goal, only changes to the JOOD schedule will be emphasized in this analysis. Changes in the BDOC schedule would lower overall wait time for that specific node of training, but it would not lower the overall time spent in the system, because the entity would still have to wait for JOOD. Instead, it would raise the wait time at a different node of training. The schedules for all the JOOD locations were compressed in order to accommodate more classes throughout the year. If compressed, JOOD classes at Norfolk, San Diego, and Newport were all able to allow one more class throughout the year each.

Compressing the schedule at JOOD has limitations. There is enough space available for classes to overlap, but there are not enough simulators. JOOD has Conning Officer Virtual Environment (COVE) stations to provide the navigation and ship handling training for the students. There are 10 COVEs available that can take three students per COVE. As a result, there can only be one class a time. However, since week 1 of class is primarily classroom, while the other three weeks are primarily spent in the COVE, classes can overlap one week maximum. This is the case for JOOD classes in Norfolk and San Diego. No overlaps were done for JOOD classes in Newport since the classroom capacity is lower, therefore classroom size does not permit the same overlap luxury as the other JOOD locations.

The model tested six different schedule compression scenarios. Given that there are three JOOD locations, the scenarios were based off different combinations of schedules that were compressed. The scenarios are outlined as follows:

1. Original scheduling

2. Norfolk or San Diego schedule compressed

3. Newport schedule compressed

4. Norfolk and San Diego schedule compressed

5. Norfolk and Newport schedule compressed

6. All JOOD locations compressed

Compressing only the San Diego schedule would result in the same conclusion as only compressing Norfolk since they have the exact same schedule and the model logic of sending entities to the first available JOOD regardless of geographic location. Therefore, that scenario was not run. The scenario of compression San Diego and Newport was also not run for the same reason. Figure 25 outlines the wait time change results for each scenario.

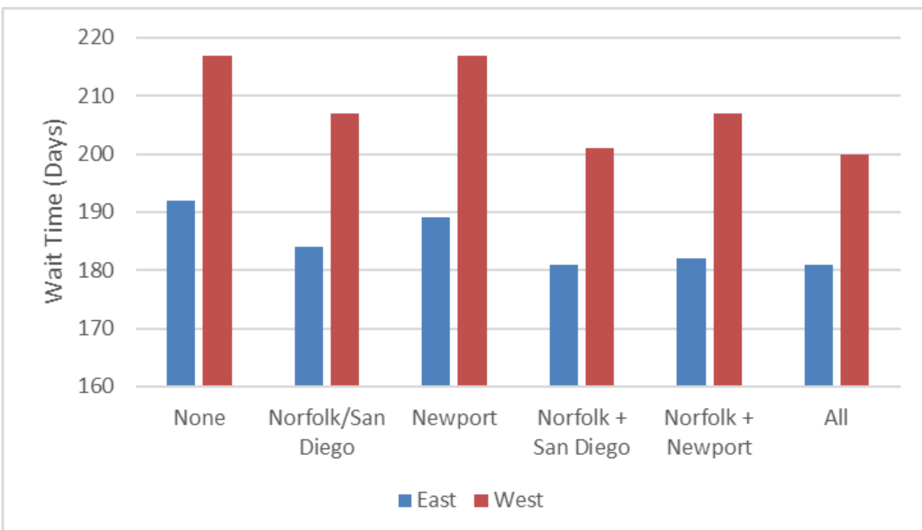


Figure 25. Mean wait time vs. schedule compressed

The first scenario shows the baseline mean wait time associated with the "Do nothing" case. In order to see how these cases compare to each other, Figure 26 outlines the average amount of time saved. Mean time saved is shown for East commissions, West commissions, and overall. Overall is the sum of East and West.

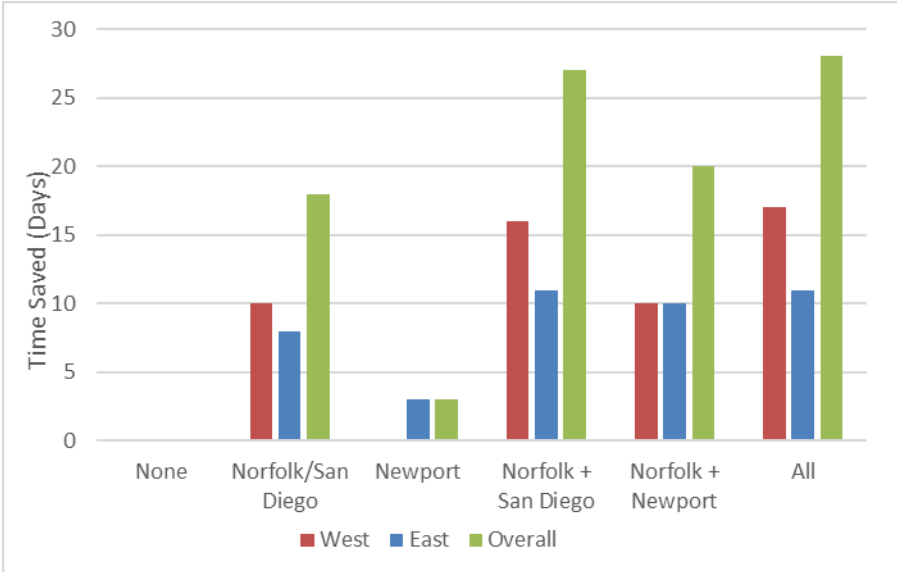


Figure 26. Time saved vs. schedule compressed

Figure 26 shows that the first large return in time saved would be to compress the schedule in Norfolk or San Diego, while compressing only Newport would only yield a three-day difference. If two schedules are compressed, the preferred schedules would be from Norfolk and San Diego. The best-case scenario would be to compress all three schedules, which would save an overall time of 28 days between East and West commissions.

Another benefit of schedule compression is the number of SWO officers that can complete training each year would rise. The model assumes that the number of accessions is the same every year. Therefore, the only benefit of schedule compression as seen in this analysis is the ability to complete the training for the same number of SWOs each year in a shorter period. The number of empty, or underutilized classrooms would also increase if this remained the case. In reality, the number of commissions does not remain constant each year. If the number of commissions were to rise, which is expected, this schedule compression can allow for the complete training of up to 84 SWOs per year. Another metric of interest is the average amount of man-days saved in one year of simulation. If an average of 1,008 students complete training 28 days sooner than the original schedule, this corresponds to 28,224 man-days saved, or just over 77 man-years of friction reduced in one year.

C. PYTHON COMPARISON

The Simio program provides all the previously mentioned results outlined in this chapter. Due to time constraints, Python has not yet been employed for statistical analysis. The foundation for the Python model has been created and can take overall statistics like the Simio model, but it cannot take monthly or class metrics. The overall statistics can still test the overall validity and functionality of the Python model. The following table shows a comparison of the numbers achieved via Simio and Python. These numbers are based off a single run with no warmup period.

Table 4. BDOC averages (days)

Region	Number in Queue		Time in Queue	
	Python	Simio	Python	Simio
East	70.098	57.623	63.334	50.847
West	120.656	112.509	64.421	68.373

Table 5. JOOD averages (days)

Region	Number in Queue		Time in Queue	
	Python	Simio	Python	Simio
East	68.088	53.074	66.535	54.058
West	76.849	72.771	70.523	54.761

The numbers provided in Tables 4 and 5 are relatively close to one another, especially considering this is a single run simulation instead of multiple experiments. The manner in which the statistics are collected are different with each program, therefore more specific statistics such as monthly metrics are not available. Differences in these numbers can be explained via different number generators between the programs and how entities are transferred from minor logical differences. The logical process in Simio is dictated with pictured Boolean blocks. The processes are hardcoded in Python, therefore there may be some logical differences that are not accounted for.

D. VALIDATION/VERIFICATION

The process of verifying both models involved running simplified versions of the model. For example, the Simio and Python models were tested with a simple M/M/1 queues to ensure the simulation matched analytic results. Animation graphs also verified the model in Simio while validating similar behavior to the real world. The Python model also shared close results to the Simio model, ensuring that two different implementations of the SWO model closely matched. Subject matter experts in PERS-41 were the primary validation sources for the models.

E. SUMMARY

This chapter outlined the results from a single run simulation as well as a stochastic analysis from 50 experiments in Simio. The analysis begun with an exploration of steady state statistics with the following metrics: number of students waiting, average time waiting per student, man-days spent waiting for training per month, and class utilization rates per class. The exploration then pivoted to a sensitivity analysis where schedule compression was the main change. Other changes to the model, such as an extra instructor or larger classrooms, provided minimal change to the analysis and were also beyond the control for the training commands at these locations. The chapter ended with a comparison with what statistics the Python model can gather.

Chapter V discusses the conclusion and recommendations based on the analysis from this chapter followed by a future work section for potential theses. The future work section opens with the progress made in the Python model and potential improvements that can be made to the model. It ends with a discussion of existing training evolutions in the SWO training pipeline that is of interest in evaluating to PERS 41 leadership.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSION

This analysis assumes steady state conditions, it also assumes there is no backlog of SWOs that are currently waiting for JOOD training. Because the first JOOD class concluded in July of 2019, this is likely not the case. If this backlog exists, then every class will be utilized to its maximum degree, meaning a compression of either Norfolk or San Diego is the best possible option. This model also is subject to change with the full implementation of JOOD training in 2021. Currently, the course is four weeks long. By the time it is fully implemented, it will transition to a six-week course (Department of the Navy 2018). This will lower the number of JOOD courses that commence each year, further escalating the need for schedule compression.

A. RECOMMENDATION

The current schedule does not optimize the number of SWOs that complete BDOC and JOOD training each year. It also does not graduate the number of SWOs from these trainings as efficiently as possible. A maximum of 1,164 SWOs can complete these trainings each year. The current system can be improved by compressing the schedule for JOOD in either Norfolk or San Diego. This would raise the maximum number of SWOs that can complete BDOC and JOOD each year to 1,194 SWOs. This would also lower the average amount of friction experienced by SWOs by 18 ± 3 days. If both Norfolk and San Diego were compressed, a maximum of 1,224 SWOs can complete training each year while reducing friction by 27 ± 3 days.

Something to consider is that there are five classes per year across the three JOOD locations where the classrooms are either empty or at least underutilized, as discussed in Chapter IV. This means that the maximum number of SWOs that can complete these trainings in steady state conditions will not happen. Under these circumstances, the best possible time saving scenario would be to compress the schedules at all locations. This will lower the average amount time to complete training experienced by 28 ± 3 days. The best possible option will depend on the number of SWOs backlogged due to the high commission rate in May each year.

B. FUTURE WORK

Continuing work on this thesis can be broken down into two categories: Improvements on the existing DIVO model, and future models that can be creating using the DESpy package in Python. Improvements to the DIVO model are necessary to better compare the model to Simio and test improvements to the SWO training schedule at this stage of training. This model is also the beginning of the entire SWO career path and once other training stages are modeled, observing the interrelationship can provide long term effects to minor changes to the training pipeline.

Concepts from the DIVO model can also be translated into other models for analysis. The SWO career path outlined in Figure 1 spans roughly 26 years. The DIVO model is responsible for about one year. There are other training sectors in the pipeline, such as ADOC and BST after the initial DIVO tour and DH training after a SWO's shore tour. The DH training program is of interest for the leadership at PERS 41 and will likely be next section modelled.

1. The DIVO Model

The DIVO model in Python has been created and provides numbers that are within reason in comparison to Simio. However, the majority of this thesis was spent creating and evaluating the Simio model. Python became a consideration late in the project due to reevaluating the practical uses of Simio for leadership in Millington, TN. The manner in which statistics are gathered in DESpy require improvement. As of now, the only metrics that are measured are average queue sizes, time in queue, and average class sizes. These are not broken down by month or class and are only provided as overall averages throughout the entire simulation. This is insufficient for assessing changes to the training schedule to graduate the optimal amount of SWOs prior to their first DIVO tour. Updating the program to provide desired statistics for PERS 41, creating a reasonable number of simulations for stochastic analysis, and organizing it in a user-friendly manner for nonprogrammers are just a few tasks that would provide great thesis opportunities for future students. Another area of improvement would be the visualization aspect. Visualization comes easy to Simio and the program can create several graphs depicting the behavior of the entities very easily. Images depicting entity behavior will be more difficult in Python but would be very helpful for analysis.

2. Future Models

The next model of interest is the DH model. The training requirements for specific billets must be completed prior to a SWO reporting to their ultimate duty station. PERS-41 has several challenges when scheduling DH training such as ship schedule, officer timing, course timing, and available quotas that affect training tracks (Department of the Navy 2019). This creates a strong need to model the DH training pipeline to observe outcomes of potential changes. The consequences due to friction at the DH stage of the SWO pipeline are greater than the initial DIVO training state and is the logical next model for future thesis work.

The DH model is more complicated than the DIVO model and is difficult to evaluate. This is due to the specificity of the training. In the DIVO model, all SWOs can attend any BDOC class and any JOOD class, as the training is the same regardless of location. In the DH model, after every SWO attends DH school, they are transferred to both billet specific training courses and platform specific training (PST) courses. This creates a complex network where several SWOs of different platforms and billets can separate or overlap depending on their duty station. If the proposed changes and updates are incorporated into the Python DIVO model, then the only future students need only focus on the logic updates for the DH model.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX. PYTHON LOGIC

The DIVO model created in Python requires specific logic when sending entities from a Western or Eastern queue to a class. For the BDOC classes, there are two queues to grab entities from and two classes to send them too. For JOOD, there are two queues and three classes to send entities to. Each section in this appendix provides the logic and the pseudocode that the model uses. Images that illustrate the pseudocode from a BDOC and JOOD perspective are provided. The following variables are used in the logic throughout this section:

cap: capacity or class size

Q: number of entities in queue

section: the number of entities taken from Eastern/Western queues

cohort: the entities scheduled for the next class

The Arr event adds to the Eastern/Western queues and the BDOC or JOOD events are associated with a class size. The goal of the logic is to determine how many entities from each queue to send to class.

A. BDOC LOGIC

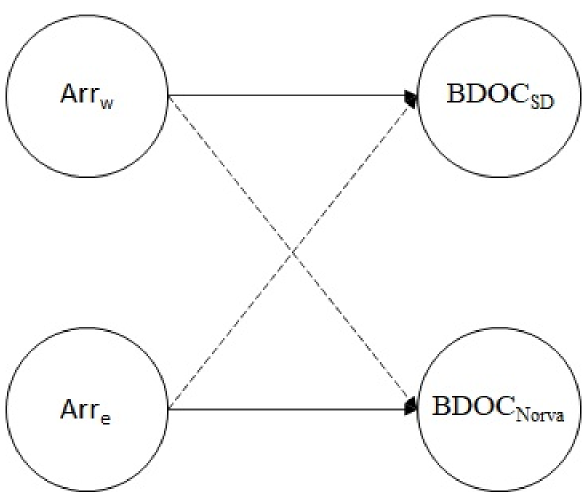


Figure 27. BDOC logic illustration

1. San Diego Perspective

if $cap_w > Q_w$ **and** $Q_e == 0$:

$cohort_w = Q_w$

else if $cap_w \geq Q_w$ **and** $cap_e \geq Q_e$ **and** $Q_e \geq cap_w - Q_w$

east_section = $cap_w - Q_w$

west_section = Q_w

$cohort_w$ = east_section + west_section

else if $cap_w \geq Q_w$ **and** $cap_e \geq Q_e$ **and** $Q_e < cap_w - Q_w$

east_section = Q_e

west_section = Q_w

$cohort_w$ = east_section + west_section

else if $cap_w \leq Q_w$:

$cohort_w = cap_w$

else if $cap_w > Q_w$ **and** $cap_e < Q_e$

east_section = $cap_w - Q_w$

west_section = Q_w

$cohort_w$ = west_section + east_section

2. Norfolk Perspective

if $cap_e > Q_e$ **and** $Q_w == 0$:

$cohort_e = Q_e$

else if $cap_e \geq Q_e$ **and** $cap_w \geq Q_w$ **and** $Q_w \geq cap_e - Q_e$

west_section = $cap_e - Q_e$

east_section = Q_e

$cohort_e$ = east_section + west_section

else if $cap_e \geq Q_e$ **and** $cap_w \geq Q_w$ **and** $Q_w < cap_e - Q_e$

east_section = Q_e

west_section = Q_w

$cohort_e$ = east_section + west_section

else if $cap_e \leq Q_e$:

$cohort_e = cap_e$

else if $cap_e > Q_e$ **and** $cap_w < Q_w$

$west\_section = cap_e - Q_e$

$east\_section = Q_e$

$cohort_e = west\_section + east\_section$

B. JOOD LOGIC

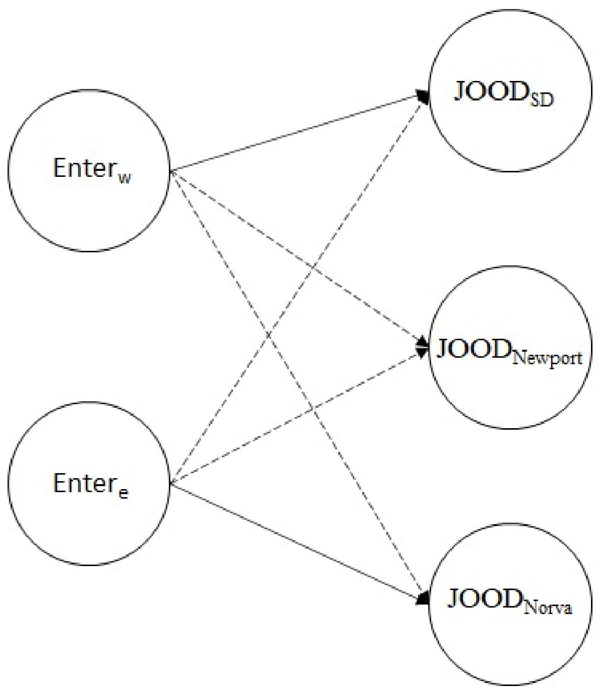


Figure 28. JOOD logic illustration

1. San Diego Perspective

if $cap_{sd} > Q_w$ **and** $Q_e == 0$:

$cohort_{sd} = Q_w$

else if $cap_{sd} \geq Q_w$ **and** $cap_{norva} \geq Q_e$ **and** $Q_e \geq cap_{sd} - Q_w$

$east\_section = cap_{sd} - Q_w$

$west\_section = Q_w$

$cohort_{sd}$ = east_section + west_section

else if $cap_{sd} \geq Q_w$ **and** $cap_{norva} \geq Q_e$ **and** $Q_e < cap_{sd} - Q_w$

east_section = Q_e

west_section = Q_w

$cohort_{sd}$ = east_section + west_section

else if $cap_{sd} \leq Q_w$:

$cohort_{sd} = cap_{sd}$

else if $cap_{sd} > Q_w$ **and** $cap_{norva} < Q_e$

east_section = $cap_{sd} - Q_w$

west_section = Q_w

$cohort_{sd}$ = west_section + east_section

2. Norfolk Perspective

if $cap_{norva} > Q_e$ **and** $Q_w == 0$:

$cohort_{norva} = Q_e$

else if $cap_{norva} \geq Q_e$ **and** $cap_{sd} \geq Q_w$ **and** $Q_w \geq cap_{norva} - Q_e$

west_section = $cap_{norva} - Q_e$

east_section = Q_e

$cohort_{norva}$ = east_section + west_section

else if $cap_{norva} \geq Q_e$ **and** $cap_{sd} \geq Q_w$ **and** $Q_w < cap_{norva} - Q_e$

east_section = Q_e

west_section = Q_w

$cohort_{norva}$ = east_section + west_section

else if $cap_{norva} \leq Q_e$:

$cohort_{norva} = cap_{norva}$

else if $cap_{norva} > Q_e$ **and** $cap_{sd} < Q_w$

west_section = $cap_{norva} - Q_e$

east_section = Q_e

$cohort_{norva}$ = west_section + east_section

3. Newport Perspective

if $Q_w + Q_e \leq cap_{new}$

    west_section = Q_w

    east_section = Q_e

    $cohort_{new}$ = west_section + east_section

else if $Q_w \geq cap_{new}$ **and** $Q_e \leq cap_{new}$

    $cohort_{new} = cap_{new}$

else if $Q_e \geq cap_{new}$ **and** $Q_w \leq cap_{new}$

    $cohort_{new} = Q_e$

else if $Q_w > cap_{new}$ **and** $Q_e > cap_{new}$

  if $Q_w \geq Q_e$

    $cohort_{new} = cap_{new}$

  else

    $cohort_{new} = cap_{new}$

else if $Q_w < cap_{new}$ **and** $Q_e < cap_{new}$ **and** $Q_w + Q_e > cap_{new}$

  if $Q_w \geq Q_e$

    west_section = Q_w

    east_section = $cap_{new} - Q_w$

    $cohort_{new}$ = west_section + east_section

  else

    east_section= Q_e

    west_section = $cap_{new} - Q_e$

    $cohort_{new}$ = west_section + east_section

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Buss A (2001) Basic event graph modeling. *Simul. News Eur.* (31), http://hdl.handle.net/10945/45519.

Cancian M (2019) U.S. military forces in FY 2020: Navy. Accessed July 6, 2020, https://www.csis.org/analysis/us-military-forces-fy-2020-navy.

Commander, Naval Surface Force, U. S. Pacific Fleet (2019) Inaugural Junior Officer of the Deck Course Lays Foundation for Junior Officers to Drive Ships, Lead Sailors. Accessed Jan 22, 2020, https://www.public.navy.mil/surfor/Pages/Inaugural%20Junior%20Officer%20of%20the%20Deck%20Course%20Lays%20Foundation%20for%20Junior.aspx.

Davenport J, Neu C, Smith W, Heath S (2007) Using discrete event simulation to examine marine training at the Marine Corps communication-electronics school. *2007 Winter Simul. Conf.* (IEEE, Washington, DC). https://ieeexplore.ieee.org/abstract/document/4419747?casa_token=Uuo6Uf_1HPYAAAAA:OWWQmQcN8O2znT-jvFq9MYyduDJxQTD6gNYXx_SYBNqpxtsV6O5gsDaISa_TXOA3aVFSr6c.

Department of the Navy (2017) Comprehensive review of recent surface force incidents. Washington, DC. http://s3.amazonaws.com/CHINFO/Comprehensive+Review_Final.pdf

Department of the Navy (2017) Strategic Readiness Review. Washington, DC. https://s3.amazonaws.com/CHINFO/SRR+Final+12112017.pdf

Department of the Navy (2018) Surface Warfare Officer Qualification and Authority to wear the Insignia. COMNAVSURFPAC/COMNAVSURFLANTINST 1412.1A, Washington, DC.

Department of the Navy (2019) Billet Specialty Training for Officers en route to Surface Force Ships. COMNAVSURFOR INSTRUCTION 1211.3, Washington, DC.

Gable A (2014) Study of Naval Air Station operations to reduce fuel consumption. Master's thesis, Department of Operations Research. Naval Postgraduate School, Monterey, California. http://hdl.handle.net/10945/42629.

Gross D, Shortle JF, Thompson JM, Harris CM (2008) *Fundamentals of queueing theory* 4th ed. (Wiley, Hoboken, NJ).

Nguyen V, Novak A, Shokr M, Pash K (2017) Aircrew manpower supply modeling under change: An agent-based discrete event simulation approach. *2017 Winter Simul. Conf. WSC*. (IEEE, Las Vegas, NV). http://ieeexplore.ieee.org/document/8248116/.

Pegden DC, Sturrock DT (2013) *Rapid modeling solutions: Introduction to simulation and simio* (CreateSpace Independent Publishing Platform, Sewickley, PA).

Trinque D (2019) Surface Warfare Officer Community brief, August 02, Millington, TN. https://www.public.navy.mil/bupers-npc/officer/Detailing/surfacewarfare/Documents/Community%20Brief%20%2022AUG19.pdf.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California